Learning Binomial Tests with Python: A Step-by-Step Guide

Authored by

Mohammed looti

November 7, 2025

The **binomial test** serves as a cornerstone in statistical inference, providing a robust methodology for comparing an observed **sample proportion** against a predetermined or **hypothesized proportion**. This powerful statistical procedure is specifically tailored for scenarios involving binary data--outcomes that can be neatly classified as one of two mutually exclusive categories, typically labeled "success" or "failure."

This test is indispensable for researchers and analysts aiming to determine whether the frequency of a specific outcome, observed across a fixed number of independent trials, deviates significantly from the frequency that is theoretically expected under a standard, baseline assumption. By quantifying this deviation, the binomial test allows us to make objective decisions about population parameters based on limited sample data.

Statistical Foundations: When and Why to Use the Binomial Test

The utility of the binomial test rests upon the principles of the binomial distribution, which models the number of successes in a fixed number of independent **Bernoulli trials**. A classic example to illustrate its application involves evaluating the fairness of a simple device, such as a standard six-sided die, where the outcome of interest is defined as rolling a specific number, say "3."

If we assume the die is perfectly fair, the theoretical probability (P) of observing the number "3" on any single roll is precisely 1/6, or approximately 0.1667. If we decide to conduct 12 independent rolls, our statistical expectation would be to observe the number "3" approximately 12 multiplied by (1/6), leading to an expected frequency of 2 occurrences. This expected frequency forms the basis of our statistical comparison.

However, experimental data rarely aligns perfectly with theoretical expectations. Suppose our actual observation shows that the number "3" appears 4 times out of those 12 rolls. This observed frequency of 4 represents a notable discrepancy from the expected 2. The critical statistical challenge posed by this observation is discerning whether this variation is merely a product of inherent random fluctuation--a natural variability expected in any sampling process--or if it is evidence suggesting the die is genuinely **biased** towards that specific outcome. The binomial test provides the rigorous mathematical framework necessary to calculate the likelihood of observing such an extreme result (or more extreme) purely by chance, assuming the initial baseline assumption remains true.

Implementing the Binomial Test in Python

For modern data analysis and robust statistical computation, the Python ecosystem offers efficient and highly accessible tools for performing rigorous **hypothesis testing**. The primary resource for this task is the `binom_test` function, which is a core component of the specialized **scipy.stats** library--the foundational package for scientific computing in Python.

Utilizing this function significantly streamlines the analytical workflow. Instead of requiring manual calculations using complex probability mass functions, the `binom_test` function directly calculates and returns the crucial **p-value**, which is the cornerstone for evidence-based decision-making in any hypothesis test. Before any computation can begin, the function must be imported into the current Python environment, typically using the following syntax:

from scipy.stats import binom_test

The standard syntax for invoking this powerful function is structured clearly, requiring several essential parameters to accurately define the scope and nature of the statistical inquiry:

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

A precise understanding of these arguments is vital for accurate test execution:

x: Represents the specific number of observed "successes" that were counted within the collected sample data.
n: Defines the total number of independent trials or observations that were conducted during the experiment.
p: This is the hypothesized proportion, representing the theoretical probability of success on any single trial, a value derived directly from the conditions of the **null hypothesis** (H0).
alternative: This parameter specifies the nature of the alternative hypothesis (HA). The default setting, 'two-sided', tests for any deviation (either greater or less) from the hypothesized proportion. However, users frequently specify 'greater' or 'less' when conducting directional **one-tailed tests**.

Interpreting Results: The Role of the P-Value

The singular output of the `binom_test` function is the **p-value**. In statistical terms, the p-value quantifies the probability of obtaining test results at least as extreme as the results actually observed, assuming that the **null hypothesis** is true. This value acts as the primary metric for evidence-based decision-making in hypothesis testing.

To draw a formal conclusion, the calculated p-value must be rigorously compared against a predetermined **significance level**, commonly denoted as alpha (α). The standard convention in most scientific and analytical fields sets this threshold at 0.05 (or 5%). The interpretation follows a strict decision rule:

If the P-value is less than or equal to α ($P \leq 0.05$), the result is deemed statistically significant. We possess sufficient evidence to reject the null hypothesis, concluding that the observed data is highly unlikely to have occurred by chance alone.
If the P-value is greater than α ($P > 0.05$), we fail to reject the null hypothesis. This means the

observed data could reasonably have occurred even if the null hypothesis were true, and thus, we lack sufficient evidence to support the alternative hypothesis.

The subsequent case studies will demonstrate the practical application of this function and illustrate how to structure the input parameters and accurately interpret the resulting p-values across diverse analytical scenarios.

Case Study 1: Analyzing Potential Die Bias

In this initial scenario, our objective is to determine empirically if a standard six-sided die exhibits any statistical bias towards the number "3." To gather the necessary data, we execute a total of 24 independent rolls (n=24) and meticulously record that the number "3" appears exactly 6 times (x=6). Given that the expected theoretical proportion (π) for a fair die is 1/6, we are specifically interested in testing whether the die is biased **towards** this number--meaning the true proportion is greater than 1/6. This directional interest necessitates the use of a **one-tailed test**.

We must formalize our statistical hypotheses based on the population proportion (π):

H0 (Null Hypothesis): $\pi \leq 1/6$ (The die is not statistically biased towards the number "3." The observed results are due to random chance.)

HA (Alternative Hypothesis): $\pi > 1/6$ (The die is statistically biased towards the number "3.")

To execute this specific one-tailed test in Python, we supply the function with the observed successes (x=6), the total trials (n=24), the null probability (p=1/6), and critically, we specify the direction of the alternative hypothesis using the argument `alternative='greater'`.

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

0.1995295129479586

The resulting **p-value** is calculated to be approximately 0.1995. When we compare this value to our conventional alpha level ($\alpha = 0.05$), we observe that P-value > α (0.1995 > 0.05). Based on this quantitative evidence, we must conclude that we do not have sufficient statistical proof to warrant the rejection of the **null hypothesis**. Therefore, we conclude that the observed frequency of 6 rolls of "3" out of 24 is likely attributable to random chance, and not indicative of a genuine, statistically significant bias.

Case Study 2: Assessing Coin Fairness

Our second practical example involves scrutinizing the fairness of a standard coin. We conduct a

series of 30 independent flips (n=30) and record that the coin lands on heads 19 times (x=19). If we assume the coin is truly fair, the expected theoretical probability (π) of landing on heads would be precisely 1/2 (0.5). Our goal is to test whether the observed data suggests that the coin is biased specifically toward yielding heads.

Since the question focuses on whether the observed proportion is significantly higher than the expected proportion, we utilize another one-sided test (a right-tailed test). The formalized hypotheses guiding our analysis are:

H0 (Null Hypothesis): $\pi \leq 1/2$ (The coin is not biased towards heads; the true probability is 0.5 or less.)

HA (Alternative Hypothesis): $\pi > 1/2$ (The coin is statistically biased towards heads; the true probability is greater than 0.5.)

We execute the `binom_test` function, providing the input parameters: 19 observed successes out of 30 trials, with the expected probability set at p=1/2, and maintaining the directional alternative hypothesis:

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

0.10024421103298661

The resulting **p-value** is calculated to be approximately 0.10024. In the context of our decision rule, we compare this P-value against the established alpha level of 0.05. Because 0.10024 > 0.05, we must once again fail to reject the null hypothesis.

Although the observed proportion (19/30 ≈ 0.633) is noticeably higher than the expected 0.5, the difference is not sufficiently large to be considered statistically significant at the standard 5% threshold. We conclude that this observation does not provide strong enough evidence to confidently claim that the coin is truly biased, suggesting that 19 heads in 30 flips is a reasonably common occurrence even with a fair coin.

Case Study 3: Evaluating New System Effectiveness

In this final example, we transition to an industrial application involving quality control. A manufacturing facility operates with a historically documented effectiveness rate of 80% (π=0.80) for its core product, the widgets. The management implements a new production system, explicitly aiming to significantly improve this baseline effectiveness rate. To evaluate the system's impact, a quality control team selects a random sample of 50 widgets (n=50) from a recent run, identifying that 47 of them are effective (x=47).

The central statistical question is whether this impressive observed increase (47 out of 50, or 94%) provides statistically significant evidence that the new system has genuinely achieved a higher effectiveness rate compared to the historical baseline of 0.80. As we are only interested in detecting improvement (an increase), this requires a directional **one-tailed test** focused on the 'greater' alternative.

The formalized hypotheses necessary for this manufacturing assessment are:

H0 (Null Hypothesis): $\pi \leq 0.80$ (The new system does not lead to a statistically significant increase in effectiveness compared to the historical rate.)

HA (Alternative Hypothesis): $\pi > 0.80$ (The new system significantly improves effectiveness, pushing the true rate above 80%.)

We execute the test in Python, inputting the observed successes (x=47), the total trials (n=50), and the historical probability (p=0.8) as the basis of the null hypothesis:

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

0.005656361012155314

The calculated **p-value** is approximately 0.00565. Since this value is considerably less than the conventional significance threshold ($\alpha = 0.05$), we possess robust statistical grounds to confidently reject the **null hypothesis**.

This strong statistical evidence confirms that the observed increase in effectiveness is highly unlikely to be the result of random sampling variability. Consequently, we can definitively conclude that the implementation of the new production system has led to a statistically significant and measurable increase in the effectiveness rate of the widgets, thereby confirming the efficacy and success of the system change.