How can hypothesis testing be performed in Python, and what are some examples of its application?

Authored by

stats writer

June 28, 2024

Hypothesis testing is a statistical method used to determine if a certain hypothesis or claim about a population is true or not. In Python, hypothesis testing can be performed using various statistical packages such as SciPy and Statsmodels. These packages provide functions and methods for conducting different types of hypothesis tests, such as t-tests, ANOVA, and chi-square tests.

To perform hypothesis testing in Python, one must first define the null and alternative hypotheses and select an appropriate test based on the type of data and research question. The necessary data must then be imported into the Python environment and the chosen test function can be applied to the data. The test results will provide a p-value, which is used to determine the statistical significance of the hypothesis.

Some examples of applications of hypothesis testing in Python include determining if there is a significant difference in sales between two versions of a product, testing the effectiveness of a new marketing strategy, and analyzing the impact of a certain variable on a population. Hypothesis testing in Python allows for efficient and accurate analysis of data, making it a valuable tool in various fields such as business, healthcare, and social sciences.

Perform Hypothesis Testing in Python (With Examples)

A is a formal statistical test we use to reject or fail to reject some statistical hypothesis.

This tutorial explains how to perform the following hypothesis tests in Python:

One sample t-testTwo sample t-testPaired samples t-test

Let's jump in!

Example 1: One Sample t-test in Python

A is used to test whether or not the mean of a

population is equal to some value.

For example, suppose we want to know whether or not the mean weight of a certain species of some turtle is equal to 310 pounds.

To test this, we go out and collect a simple random sample of turtles with the following weights:

Weights: 300, 315, 320, 311, 314, 309, 300, 308, 305, 303, 305, 301, 303

The following code shows how to use the ttest_1samp() function from the scipy.stats library to perform a one sample t-test:

```
import scipy.stats as stats

#define data
data = #perform one sample t-test
stats.ttest_1samp(a=data, popmean=310)
```

```
Ttest_1sampResult(statistic=-1.5848116313861254,
pvalue=0.138994427515873)
```

The t test statistic is -1.5848 and the corresponding two-

sided p-value is 0.1389.

The two hypotheses for this particular one sample t-test are as follows:

H0: μ = 310 (the mean weight for this species of turtle is 310 pounds)HA: μ ? 310 (the mean weight is *not* 310 pounds)

Because the p-value of our test (0.1389) is greater than alpha = 0.05, we fail to reject the null hypothesis of the test.

We do not have sufficient evidence to say that the mean weight for this particular species of turtle is different from 310 pounds.

Example 2: Two Sample t-test in Python

For example, suppose we want to know whether or not the mean weight between two different species of turtles is equal.

To test this, we collect a simple random sample of turtles from each species with the following weights:

Sample 1: 300, 315, 320, 311, 314, 309, 300, 308, 305,

303, 305, 301, 303

Sample 2: 335, 329, 322, 321, 324, 319, 304, 308, 305, 311, 307, 300, 305

The following code shows how to use the function from the scipy.stats library to perform this two sample t-test:

```
import scipy.statsas stats

#define array of turtle weights for each sample
sample1 = sample2 =

#perform two sample t-test
stats.ttest_ind(a=sample1, b=sample2)
```

```
Ttest_indResult(statistic=-2.1009029257555696, pvalue=0.0463501389516516)
```

The t test statistic is -2.1009 and the corresponding two-sided p-value is 0.0463.

The two hypotheses for this particular two sample t-test are as follows:

H0: μ1 = μ2 (the mean weight between the two species

is equal)HA: μ1 ? μ2 (the mean weight between the two species is not equal)

Since the p-value of the test (0.0463) is less than .05, we reject the null hypothesis.

This means we have sufficient evidence to say that the mean weight between the two species is not equal.

Example 3: Paired Samples t-test in Python

A is used to compare the means of two samples when each observation in one sample can be paired with an observation in the other sample.

For example, suppose we want to know whether or not a certain training program is able to increase the max vertical jump (in inches) of basketball players.

To test this, we may recruit a simple random sample of 12 college basketball players and measure each of their max vertical jumps. Then, we may have each player use the training program for one month and then measure their max vertical jump again at the end of the month.

The following data shows the max jump height (in

inches) before and after using the training program for each player:

Before: 22, 24, 20, 19, 19, 20, 22, 25, 24, 23, 22, 21

After: 23, 25, 20, 24, 18, 22, 23, 28, 24, 25, 24, 20

The following code shows how to use the from the scipy.stats library to perform this paired samples t-test:

```
import scipy.stats as stats

#define before and after max jump heights
before = after =

#perform paired samples t-test
stats.ttest_rel(a=before, b=after)
```

```
Ttest_relResult(statistic=-2.5289026942943655,
pvalue=0.02802807458682508)
```

The t test statistic is -2.5289 and the corresponding two-sided p-value is 0.0280.

The two hypotheses for this particular paired samples t-test are as follows:

H0: μ1 = μ2 (the mean jump height before and after using the program is equal)HA: μ1 ? μ2 (the mean jump height before and after using the program is not equal)

Since the p-value of the test (0.0280) is less than .05, we reject the null hypothesis.

This means we have sufficient evidence to say that the mean jump height before and after using the training program is not equal.

Additional Resources

You can use the following online calculators to automatically perform various t-tests: