

How do you perform a correlation test in Python?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *How do you perform a correlation test in Python?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=157350>

A correlation test in Python is a statistical method used to measure the strength and direction of the relationship between two variables. To perform a correlation test in Python, the following steps can be followed:

1. Import the necessary libraries: The first step is to import the required libraries such as Pandas, Numpy, and Scipy.
2. Create a dataset: The next step is to create a dataset with the two variables that need to be tested for correlation.
3. Calculate the correlation coefficient: Use the built-in function "corrcoef" from the Numpy library to calculate the correlation coefficient between the two variables.
4. Check for significance: To determine if the correlation is significant, the p-value can be calculated using the "pearsonr" function from the Scipy library.
5. Interpret the results: Based on the correlation coefficient and p-value, the strength and direction of the relationship between the two variables can be interpreted.

In conclusion, performing a correlation test in Python involves importing necessary libraries, creating a dataset, calculating the correlation coefficient, checking for significance, and interpreting the results to understand the relationship between two variables.

Perform a Correlation Test in Python (With Example)

One way to quantify the relationship between two variables is to use the Pearson correlation coefficient, which measures the linear association between two variables.

It always takes on a value between -1 and 1 where:

**-1 indicates a perfectly negative linear correlation
0 indicates no linear correlation
1 indicates a perfectly**

positive linear correlation

To determine if a correlation coefficient is statistically significant, you can calculate the corresponding t-score and p-value.

The formula to calculate the t-score of a correlation coefficient (r) is:

$$t = r * \sqrt{n-2} / \sqrt{1-r^2}$$

The p-value is then calculated as the corresponding two-sided p-value for the t-distribution with n-2 degrees of freedom.

Example: Correlation Test in Python

To determine if the correlation coefficient between two variables is statistically significant, you can perform a correlation test in Python using the `pearsonr` function from the SciPy library.

This function returns the correlation coefficient between two variables along with the two-tailed p-value.

For example, suppose we have the following two arrays in Python:

```
#create two arrays
```

```
x =
```

```
y =
```

We can import the pearsonr function and calculate the Pearson correlation coefficient between the two arrays:

```
from scipy.stats.stats import pearsonr
```

```
#calculation correlation coefficient and p-value between  
x and y
```

```
pearsonr(x, y)
```

```
(0.8076177030748631, 0.004717255828132089)
```

Here's how to interpret the output:

Pearson correlation coefficient (r): 0.8076 Two-tailed p-value: 0.0047

Since the correlation coefficient is close to 1, this tells us that there is a strong positive association between the two variables.

Note that we can also extract the individual correlation

coefficient and p-value from the pearsonr function as well:

```
#extract correlation coefficient (rounded to 4 decimal places)
```

```
r = round(pearsonr(x, y), 4)
```

```
print(r)
```

```
0.8076
```

```
#extract p-value (rounded to 4 decimal places)
```

```
p = round(pearsonr(x, y), 4)
```

```
print(p)
```

```
0.0047
```

These values are a bit easier to read compared to the output from the original pearsonr function.

Additional Resources

The following tutorials provide additional information about correlation coefficients:

[An Introduction to the Pearson Correlation Coefficient](#)