

How to calculate Spearman Rank Correlation in Python

Authored by
stats writer

December 17, 2025

RECOMMENDED CITATION

stats writer (2025). *How to calculate Spearman Rank Correlation in Python*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=107653>

Understanding Correlation in Statistical Analysis

In the realm of statistics and data science, assessing the relationship between different datasets is a fundamental requirement. The concept of **correlation** quantifies both the strength and the directionality of the linear or monotonic relationship existing between two measured variables. Understanding this measure allows analysts to determine if changes in one variable reliably correspond to changes in another, serving as a critical step in predictive modeling and causal inference studies.

The mathematical output of a correlation calculation, known as the correlation coefficient, is always a value constrained within the range of -1 to 1. This standardized scale provides an immediate interpretation of the relationship observed in the data. A coefficient near zero suggests little to no interdependence, while values closer to the extremes of -1 or 1 indicate a strong, predictable relationship.

The interpretation of the correlation coefficient is crucial for translating statistical output into meaningful insight:

-1: This signifies a perfect negative relationship, meaning that as the values of one variable increase, the values of the second variable decrease proportionally and consistently.

0: This indicates that there is no identifiable linear or monotonic relationship between the two variables. They move independently of each other.

1: This represents a perfect positive relationship, where an increase in one variable is matched by a proportional and consistent increase in the other variable.

The Need for Non-Parametric Measures: Introducing Spearman's Rho

While the widely known Pearson correlation coefficient measures the strength of a linear relationship between two variables, it relies heavily on the assumption that the data is normally distributed and that the relationship is strictly linear. When these assumptions are violated, or when dealing with ordinal data (data based on ranks rather than raw values), a more robust, non-parametric approach is necessary.

This is where the **Spearman Rank Correlation** coefficient, often denoted as ρ (rho), proves invaluable. Spearman's correlation does not assess the relationship between the raw scores themselves, but rather the relationship between the ranks of those scores. It specifically measures the strength and direction of the monotonic relationship between the paired data. A monotonic relationship is one that, while not necessarily linear, consistently moves in one direction (either increasing or decreasing).

Spearman's Rho is calculated by first ranking the data for each variable separately and then applying the standard Pearson correlation formula to these ranks. This methodology makes it an ideal tool for scenarios involving ordinal data, such as comparing the ranking of student performance across two subjects, or assessing how expert judges' ratings correlate across various criteria.

Interpreting the Spearman Rank Correlation Coefficient

The interpretation of the Spearman coefficient follows the same scale logic as Pearson's coefficient, ranging from -1 to 1. However, because it operates on ranks, a high positive correlation (close to +1) indicates that high ranks in Variable 1 correspond consistently to high ranks in Variable 2, and vice versa. Conversely, a strong negative correlation (close to -1) suggests that high ranks in Variable 1 correspond consistently to low ranks in Variable 2.

A crucial advantage of using Spearman's rank correlation is its resistance to outliers, as extreme values only affect their rank marginally compared to their effect on raw scores. Furthermore, it is suitable for measuring correlation even if the relationship is curvilinear, as long as it remains consistently monotonic.

In practical terms, a correlation coefficient magnitude between 0.0 and 0.2 typically suggests a negligible relationship, 0.2 to 0.4 indicates a weak relationship, 0.4 to 0.6 signifies a moderate relationship, and 0.6 to 0.8 represents a strong relationship. Coefficients exceeding 0.8 are considered very strong. It is important to remember that correlation does not imply causation; it merely quantifies the observed interdependence between the two sets of ranked observations.

Setting Up the Python Environment and Data

To perform sophisticated statistical analysis, the Python ecosystem provides robust libraries. For handling structured data, the **Pandas** library is essential, allowing us to manage our variables within a tabular format called a DataFrame. For the calculation of the Spearman coefficient specifically, we rely on the statistical capabilities provided by the **SciPy** library, particularly the scipy.stats module.

We will illustrate this process using a simulated dataset representing the academic performance of students. Suppose we have recorded the math exam score and the science exam score for ten students in a class. Our objective is to determine if a student who ranks highly in math tends to also rank highly in science, regardless of the precise score differences.

The initial step requires importing the necessary library and defining our dataset structure. We use a Pandas DataFrame to encapsulate this data, associating each student with their respective scores. This organization ensures that the data is properly paired for the subsequent correlation

analysis.

Defining the Example Dataset in Python

The following Python code snippet demonstrates how to import Pandas and construct the DataFrame containing the scores for our 10 students (A through J). This DataFrame serves as the foundational structure upon which the Spearman Rank Correlation will be calculated.

```
import pandas as pd
```

```
#create DataFrame
df = pd.DataFrame({'student': ,
'math': ,
'science': })
```

This defined structure ensures that the scores for Math and Science are correctly linked for each student, allowing the statistical function to accurately compare the relative rankings between the two subjects.

Calculating Spearman Correlation using SciPy

Once the data is prepared, the calculation of the Spearman rank correlation in Python is streamlined through the use of the `spearmanr()` function, which resides within the `scipy.stats` module. This function is highly efficient as it automatically handles the ranking of the data internally before computing the correlation coefficient (ρ).

The `spearmanr()` function requires at minimum two arguments, representing the two variables (columns) between which the correlation is to be measured. Importantly, this function returns two values: the Spearman rank correlation coefficient (ρ) and the corresponding two-sided **p-value**. The p-value is essential for determining the statistical significance of the calculated correlation.

To execute the calculation, we must first import the specific function from `scipy.stats`. We then pass the 'math' and 'science' columns from our Pandas DataFrame to the function. The output is typically stored in two variables, often named `rho` and `p`, for clarity in subsequent analysis and interpretation.

Executing the Spearman Correlation Calculation and Output

The following code demonstrates the implementation of the `spearmanr()` function on our dataset and provides the resulting correlation coefficient and p-value.

```
from scipy.stats import spearmanr

#calculate Spearman Rank correlation and corresponding p-value
rho, p = spearmanr(df, df)

#print Spearman rank correlation and p-value
print(rho)

-0.41818181818181815

print(p)

0.22911284098281892
```

Interpreting the Statistical Output

Upon execution, we receive two primary results: the Spearman rank correlation coefficient (ρ) of **-0.41818** and the corresponding p-value of **0.22911**. These two metrics must be analyzed together to draw a statistically sound conclusion about the relationship between the math and science scores.

The value of $\rho = -0.41818$ indicates a moderate negative correlation between the two subjects. This suggests that students who tend to rank higher in math generally tend to rank slightly lower in science, and vice versa. It is not an inverse relationship, but rather a moderate tendency for the ranks to diverge.

However, the magnitude of the correlation must be evaluated alongside its p-value to determine if this observed relationship is reliable or if it likely occurred by random chance. The p-value is the probability of observing a correlation as strong (or stronger) as -0.41818 if, in reality, there was no true correlation (the null hypothesis). In most scientific disciplines, a significance level (α) of 0.05 is used as the threshold.

Since our calculated p-value (0.22911) is substantially greater than the conventional significance level of 0.05, we fail to reject the null hypothesis. Therefore, the observed moderate negative correlation is not deemed **statistically significant**. This conclusion implies that while a negative trend exists in this specific sample, we do not have sufficient evidence to conclude that this relationship holds true across the broader student population.

Advanced Extraction and Syntax Options

While assigning the output to two separate variables (ρ and p) is generally the clearest method for presentation, the `spearmanr()` function returns a single object containing both values. If an

analyst requires only one specific metric without needing the other, it is possible to index the output directly.

The correlation coefficient (ρ) is stored at index 0 of the returned object, and the p-value is stored at index 1. Utilizing this indexing method can simplify code when the statistical test is merely an intermediate step in a larger analytical pipeline where only one result is required.

The following examples demonstrate how to extract the coefficient and the p-value independently using direct indexing, providing flexibility in data handling within Python scripts:

```
#extract Spearman Rank correlation coefficient
```

```
spearmanr(df, df)
```

```
-0.41818181818181815
```

```
#extract p-value of Spearman Rank correlation coefficient
```

```
spearmanr(df, df)
```

```
0.22911284098281892
```

Conclusion and Further Resources

The Spearman Rank Correlation coefficient provides a powerful, non-parametric method for assessing monotonic relationships between variables, especially when dealing with data that is not normally distributed or consists of ordinal rankings. By utilizing the robust tools available in Python libraries like scipy.stats and Pandas, analysts can efficiently calculate and interpret this measure.

As demonstrated, the calculation itself is straightforward using the `spearmanr()` function. However, true analytical expertise lies in correctly interpreting both the correlation coefficient (ρ) and the associated p-value to determine whether the observed trend is statistically robust enough to warrant further conclusions or predictions.

Mastering Spearman's correlation ensures that researchers have the flexibility to analyze data even when the strict assumptions required by parametric tests, such as Pearson correlation, cannot be met.

For those interested in applying Spearman Rank Correlation using other statistical platforms, the following resources may be helpful:

[How to Calculate Spearman Rank Correlation in R](#)

[How to Calculate Spearman Rank Correlation in Excel](#)

[How to Calculate Spearman Rank Correlation in Stata](#)