

# How to Find the P-Value of a Correlation Coefficient in R Using `cor.test()`

Authored by  
**stats writer**

November 20, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Find the P-Value of a Correlation Coefficient in R Using `cor.test()`*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=98430>

Determining the statistical validity of a relationship between two variables is a fundamental task in data analysis. In the statistical programming environment R, the most efficient and robust method for calculating the P-value associated with a correlation coefficient is by utilizing the built-in function, `cor.test()`. This specialized function not only computes the correlation measure itself but also performs the necessary hypothesis test to assess whether the observed linear relationship is likely due to chance or if it represents a genuine association within the underlying population.

The `cor.test()` function is designed to simplify the often complex process of hypothesis testing. It requires two numeric vectors (representing the variables) as its primary arguments and produces a comprehensive output object containing the calculated test statistic, the degrees of freedom, and crucially, the resulting P-value. Understanding this output is essential for sound statistical practice. The core principle is straightforward: the magnitude of the observed correlation must be weighed against the sample size and variability. A low P-value suggests that the null hypothesis--which posits no correlation--should be rejected.

A commonly accepted threshold for establishing statistical significance is the alpha level of 0.05. When the calculated P-value falls below this threshold ( $P < 0.05$ ), researchers conclude that the variables exhibit a statistically significant linear association. Conversely, a P-value greater than 0.05 implies that the observed correlation is too weak or too variable to confidently reject the null hypothesis, suggesting that the correlation might be a result of random sampling variation. This rigorous approach ensures that data-driven conclusions are based on robust evidence rather than mere coincidence.

## The Nature of the Correlation Coefficient (r)

The correlation coefficient, typically denoted by the letter r (specifically the Pearson product-moment correlation coefficient when dealing with linear relationships between continuous data), serves as a standardized metric to quantify the strength and direction of the linear association between two variables. It is a dimensionless quantity, meaning its value is independent of the units of measurement used for the variables being analyzed. This universal property makes it an indispensable tool for comparing relationships across diverse datasets and scientific disciplines.

The range of the correlation coefficient is strictly bounded, ensuring consistency in its interpretation. It can only take on a value between **-1** and **1**, inclusive. Values close to these extremes represent stronger relationships, while values closer to zero indicate weaker or non-existent linear associations. The sign of the coefficient dictates the direction of the relationship: a positive sign means that as one variable increases, the other tends to increase, whereas a negative sign indicates an inverse relationship.

We can summarize the interpretation of the correlation coefficient based on these boundary values:

**-1:** This denotes a **perfectly negative linear correlation**. Every change in one variable is matched by an opposite and proportionate change in the other.

**0:** This signifies **no linear correlation**. The variables are independent in terms of linear association, although they might still exhibit a non-linear relationship.

**1:** This represents a **perfectly positive linear correlation**. Every increase in one variable corresponds exactly to a proportionate increase in the other.

While the magnitude of  $r$  provides insight into the practical strength of the relationship, it does not inherently confirm whether that relationship is statistically meaningful within the context of the larger population. To move beyond descriptive statistics and make inferential statements, we must proceed to calculate the corresponding test statistic--often the t-score--and the subsequent P-value.

## Theoretical Foundation: The T-Score and Hypothesis Testing

To formally test the null hypothesis ( $H_0$ : the population correlation coefficient  $\rho$  is zero) against the alternative hypothesis ( $H_a$ : the population correlation coefficient  $\rho$  is not zero), statistical software like R relies on converting the calculated sample correlation coefficient ( $r$ ) into a standardized test statistic, typically a t-score. This transformation allows us to determine how many standard errors the observed correlation is away from the hypothesized mean of zero correlation, assuming the null hypothesis is true. This process is crucial for establishing the statistical footing of our findings.

The specific formula used to calculate the t-score (or t-statistic) derived from the sample correlation coefficient ( $r$ ) and the sample size ( $n$ ) is mathematically defined as follows. This equation standardizes the correlation by accounting for the inherent variability observed in the sample data:

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

Once the t-statistic is computed, the next step involves determining the associated P-value. This P-value represents the probability of observing a sample correlation as extreme, or more extreme, than the one calculated, assuming that no correlation truly exists in the population. This probability is derived from the theoretical t-distribution. It is imperative to correctly specify the appropriate degrees of freedom (df), which for a simple bivariate correlation test is calculated as  $n-2$ , where  $n$  is the number of observations (pairs) used in the analysis. The test is typically two-sided, assessing deviation from zero in both the positive and negative directions.

While understanding this underlying statistical machinery is valuable, statistical environments like R automate these complex calculations. The primary tool for this automation, which efficiently handles all aspects of the correlation test--from calculating  $r$  to generating the t-statistic and the final P-value--is the `cor.test()` function. This function simplifies the workflow significantly,

requiring only the two variable vectors as input:

**`cor.test(x, y)`**

## Leveraging the `cor.test()` Function in R

The `cor.test()` function in R is a versatile and powerful tool within the base installation, designed specifically for performing hypothesis tests of correlation. Unlike simply using the `cor()` function, which only returns the correlation coefficient, `cor.test()` provides the full inferential statistical context required to assess statistical significance. The function defaults to calculating the Pearson product-moment correlation coefficient, which is appropriate for continuous, normally distributed data, but it can also be configured to perform non-parametric tests like Spearman's rho or Kendall's tau.

Using `cor.test()` requires minimal input, making it highly accessible even for novice R users. The syntax typically involves passing two vectors, `x` and `y`, representing the data points for the two variables under investigation. The function then executes the necessary statistical procedures internally, handling outlier checks, calculating variances, and applying the previously discussed t-statistic formula. It is important to ensure that both input vectors are of the same length and contain numerical data suitable for correlation analysis.

Beyond the simple two-variable inputs, `cor.test()` offers several optional arguments that allow for greater control over the hypothesis test. For instance, the `method` argument allows switching between "pearson", "kendall", or "spearman" methods based on the distributional assumptions of the data. Furthermore, the `alternative` argument can specify a one-sided test (e.g., "greater" or "less") if the researcher has a prior theoretical reason to expect correlation only in one direction, although the default and most common usage is the two-sided test ("two.sided"), which tests for any non-zero correlation.

The comprehensive output generated by this function provides all the necessary components for a complete statistical report, enabling the user to immediately ascertain whether the correlation is merely descriptive or if it possesses inferential power. In the following sections, we will walk through a concrete example demonstrating how to implement this function and meticulously interpret every piece of information it provides, focusing specifically on isolating and understanding the P-value.

## Practical Example 1: Assessing Correlation Significance

To demonstrate the utility of the `cor.test()` function, let us analyze a hypothetical dataset consisting of two variables, `x` and `y`. Assume `x` represents the number of hours spent studying per

week, and  $y$  represents a student's score on a standardized test. We are interested in determining if a statistically significant linear relationship exists between these two variables using the power of R.

First, we must define our data vectors within the R environment. We create ten observations for each variable and then execute the `cor.test()` function, maintaining the default Pearson method, which is appropriate for this type of continuous, interval data. The code block below illustrates the definition of the data and the execution of the primary test command:

#### **#create two variables (Hours Studied vs. Test Score)**

```
x <- c(70, 78, 90, 87, 84, 86, 91, 74, 83, 85)
```

```
y <- c(90, 94, 79, 86, 84, 83, 88, 92, 76, 75)
```

```
#calculate correlation coefficient and corresponding p-value
```

```
cor.test(x, y)
```

Executing this command generates a detailed output, providing all the necessary components for hypothesis evaluation. This output is far richer than a simple correlation matrix, offering inferential statistics that address the central question of statistical significance. The key elements presented include the chosen test type, the resulting t-statistic, the degrees of freedom (df), the essential P-value, the confidence interval, and the calculated sample correlation estimate. This comprehensive result facilitates rigorous statistical interpretation.

## **Interpreting the Output and Drawing Conclusions**

Upon reviewing the full output generated by the `cor.test(x, y)` command, we gain a detailed understanding of the relationship between the variables. The output begins by confirming the method used--in this case, Pearson's product-moment correlation. Following this, the crucial summary statistics are provided, which allow us to perform the necessary statistical inference against our null hypothesis that the true population correlation is zero.

The detailed results are presented as follows:

### **Pearson's product-moment correlation**

data: x and y

t = -1.7885, df = 8, p-value = 0.1115

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.8709830 0.1434593

sample estimates:

```
cor
-0.5344408
```

Analyzing the specific estimates extracted from this output provides two key pieces of information:

The calculated Pearson correlation coefficient ( $r$ ) is **-0.5344408**. This moderate negative value suggests that as the variable  $x$  (hours studied) increases, the variable  $y$  (test score) tends to decrease.

The corresponding test statistic is  $t = \mathbf{-1.7885}$ , calculated using  $\mathbf{8}$  degrees of freedom (df), derived from  $n-2 = 10-2 = 8$ .

The calculated P-value is **0.1115**.

The interpretation hinges entirely on comparing the P-value (0.1115) to the pre-established level of statistical significance ( $\alpha = 0.05$ ). Since  $0.1115$  is greater than  $0.05$ , we fail to reject the null hypothesis. Even though the correlation coefficient itself is moderately strong ( $-0.534$ ), this observed relationship is deemed statistically insignificant at the 5% level. This implies that while a negative linear relationship exists in this specific sample, there is insufficient statistical evidence to conclude that this correlation is present in the broader population from which the sample was drawn, suggesting the observed effect could easily be due to sampling variability or chance.

## Extracting the P-Value Directly in R

While the full output of `cor.test()` is essential for a thorough review, analysts often require only the numerical value of the P-value for use in automated scripts, tabular reporting, or conditional logic within a larger data pipeline. Fortunately, R treats the results of `cor.test()` as an object of class `htest`, meaning its components can be accessed and extracted directly using the dollar sign operator (`$`).

To specifically retrieve the P-value without displaying the full summary statistics, we simply append `$p.value` to the function call. This method is highly efficient for programmatic tasks where parsing textual output is undesirable. It returns a single numeric value, which is often displayed with greater precision than the rounded output in the console summary, ensuring maximum accuracy for subsequent calculations or reporting.

The following code block demonstrates this focused extraction, using the same variables  $x$  and  $y$  from our previous example. This technique is particularly valuable when generating automated reports or when integrating correlation tests into complex statistical models:

```
#create two variables
x <- c(70, 78, 90, 87, 84, 86, 91, 74, 83, 85)
y <- c(90, 94, 79, 86, 84, 83, 88, 92, 76, 75)
```

```
#calculate p-value for correlation between x and y  
cor.test(x, y)$p.value
```

```
0.1114995
```

The resulting output, `0.1114995`, provides the precise P-value for the test. As expected, this precise figure confirms the rounded value (0.1115) observed in the full summary output. Utilizing this extraction technique confirms the statistical finding--that the correlation is not statistically significant at the standard 0.05 level--while simultaneously providing a streamlined method for handling and processing statistical results within the R environment.

ARABPSYCHOLOGY.COM