

How to Easily Calculate Correlation Coefficients in R with cor()

Authored by
stats writer

November 22, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Calculate Correlation Coefficients in R with cor()*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=99483>

The `cor()` function in R is used to calculate correlation coefficients between two sets of data. The function takes two vectors as arguments and returns the correlation coefficient between them. This value will range from -1 to 1 and represents the linear relationship between the two datasets. Positive values indicate a positive linear relationship while negative values indicate a negative linear relationship. The closer the value is to 1 or -1, the stronger the linear relationship between the two data sets.

You can use the **cor()** function in R to calculate correlation coefficients between variables.

Here are the most common ways to use this function:

Method 1: Calculate Pearson Correlation Coefficient Between Two Variables

cor(df\$x, df\$y)

Use the Pearson correlation coefficient when calculating the correlation between two continuous variables. (e.g. height and weight)

Method 2: Calculate Pearson Correlation Coefficient Between All Numeric Variables in Data Frame

cor(df)

This method will return a matrix that contains the Pearson correlation coefficient between each pairwise combination of numeric variables in a data frame.

Method 3: Calculate Spearman Correlation Coefficient Between Two Variables

cor(df\$x, df\$y, method='spearman')

Use the Spearman correlation coefficient when calculating the correlation between two ranked variables. (e.g. rank of a student's math exam score vs. rank of their science exam score in a class)

Method 4: Calculate Kendall's Correlation Coefficient Between Two Variables

cor(df\$x, df\$y, method='kendall')

Use the Kendall correlation coefficient when when you wish to use Spearman Correlation but the sample size is small and there are many tied ranks.

The following examples show how to use each method in practice with the following data frame in R that shows the number of hours spent studying, number of practice exams taken, and final exam score for eight different students:

```
#create data frame
df <- data.frame(hours=c(1, 1, 3, 2, 4, 3, 5, 6),
  prac_exams=c(4, 3, 3, 2, 3, 2, 1, 4),
  score=c(69, 74, 74, 70, 89, 85, 99, 90))
```

```
#view data frame
```

```
df
```

```
hours prac_exams score
```

```
1 1 4 69
```

```
2 1 3 74
```

```
3 3 3 74
```

```
4 2 2 70
```

```
5 4 3 89
```

```
6 3 2 85
```

```
7 5 1 99
```

```
8 6 4 90
```

Example 1: Calculate Pearson Correlation Coefficient Between Two Variables

```
#calculate Pearson correlation coefficient between hours and score
cor(df$hours, df$score)
```

```
0.8600528
```

The Pearson correlation coefficient between **hours** and **score** turns out to be **0.86**.

Note that if there are NA values in your data frame, you can use the argument **use='complete.obs'** to only use the rows where there are no NA values:

```
#calculate Pearson correlation coefficient and ignore any rows with NA
cor(df$hours, df$score, use='complete.obs')
```

Example 2: Calculate Pearson Correlation Coefficient Between All Numeric Variables

The following code shows how to use the **cor()** function to create a correlation matrix that contains the Pearson correlation coefficient between all numeric variables in the data frame:

```
#calculate Pearson correlation coefficient between all numeric variables
```

```
cor(df)
```

```
hours prac_exams score
hours 1.0000000 -0.1336063 0.8600528
prac_exams -0.1336063 1.0000000 -0.3951028
score 0.8600528 -0.3951028 1.0000000
```

Here's how to interpret the output:

The Pearson correlation coefficient between **hours** and **prac_exams** is **-.13**.

The Pearson correlation coefficient between **hours** and **score** is **.86**.

The Pearson correlation coefficient between **prac_exams** and **score** is **-.39**.

Note: The Pearson correlation coefficient between each individual variable and itself is always 1, which is why each value along the diagonal of the correlation matrix is 1.

Example 3: Calculate Spearman Correlation Coefficient Between Two Variables

The following code shows how to use the **cor()** function to calculate the Spearman correlation coefficient between the **hours** and **prac_exams** variables:

```
#calculate Spearman correlation coefficient between hours and prac_exams
```

```
cor(df$hours, df$prac_exams, method='spearman')
```

```
-0.1250391
```

The Spearman correlation coefficient between **hours** and **prac_exams** turns out to be **-.125**.

Example 4: Calculate Kendall's Correlation Coefficient Between

Two Variables

The following code shows how to use the **cor()** function to calculate Kendall's correlation coefficient between the **hours** and **prac_exams** variables:

```
#calculate Kendall's correlation coefficient between hours and prac_exams  
cor(df$hours, df$prac_exams, method='kendall')
```

```
-0.1226791
```

Kendall's correlation coefficient between **hours** and **prac_exams** turns out to be **-.123**.

ARABPSYCHOLOGY.COM