

How can one perform a Runs Test in R?

Authored by
stats writer

April 20, 2024

RECOMMENDED CITATION

stats writer (2024). *How can one perform a Runs Test in R?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=137315>

The Runs Test is a statistical test used to determine the randomness or non-randomness of a sequence of data. In order to perform a Runs Test in R, one must first import the data into the R environment. This can be done either by manually inputting the data or by importing a data file from an external source. Once the data is in the R environment, the next step is to use the "runs.test" function, which is a built-in function in the stats package of R. This function takes the data as its input and calculates the p-value, which is used to determine the significance of the results. The result of the Runs Test can then be interpreted to determine if the data follows a random pattern or if there is a significant trend present. It is important to note that the Runs Test has certain assumptions and limitations, and it is recommended to consult with a statistician for proper interpretation and analysis of the results.

Perform Runs Test in R

Runs test is a statistical test that is used to determine whether or not a dataset comes from a random process.

The null and alternative hypotheses of the test are as follows:

H0 (null): The data was produced in a random manner.

Ha (alternative): The data was *not* produced in a random manner.

This tutorial explains two methods you can use to perform Runs test in R. Note that both methods lead to the exam same results.

Method 1: Run's Test Using the *snpar* Library

The first way you can perform Run's test is with the `runs.test()` function from the `snpar` library, which uses the following syntax:

```
runs.test(x, exact = FALSE, alternative = c("two.sided",  
"less", "greater"))
```

where:

x: A numeric vector of data values.
exact: Indicates whether an exact p-value should be calculated. This is `FALSE` by default. If the number of runs is fairly small, you can change this to `TRUE`.
alternative: Indicates the alternative hypothesis. The default is `two.sided`.

The following code shows how to perform Run's test using this function in R:

```
library(snpar)  
  
#create dataset  
data <- c(12, 16, 16, 15, 14, 18, 19, 21, 13, 13)  
  
#perform Run's test  
runs.test(data)
```

Approximate runs test

data: data

Runs = 5, p-value = 0.5023

alternative hypothesis: two.sided

The p-value of the test is 0.5023. Since this is not less than $\alpha = .05$, we fail to reject the null hypothesis. We have sufficient evidence to say that the data was produced in a random manner.

Method 2: Run's Test Using the *randtests* Library

The second way you can perform Run's test is with the `runs.test()` function from the *randtests* library, which uses the following syntax:

```
runs.test(x, alternative = c("two.sided", "less",  
"greater"))
```

where:

x: A numeric vector of data values.
alternative: Indicates the alternative hypothesis. The default is `two.sided`.

The following code shows how to perform Run's test

using this function in R:

```
library(randtests)
```

```
#create dataset
```

```
data <- c(12, 16, 16, 15, 14, 18, 19, 21, 13, 13)
```

```
#perform Run's test
```

```
runs.test(data)
```

Runs Test

data: data

statistic = -0.67082, runs = 5, n1 = 5, n2 = 5, n = 10, p-value =

0.5023

alternative hypothesis: nonrandomness

Once again the p-value of the test is 0.5023. Since this is not less than $\alpha = .05$, we fail to reject the null hypothesis. We have sufficient evidence to say that the data was produced in a random manner.