

How can the square root function be used in R and what are some examples of its implementation?

Authored by
stats writer

May 2, 2024

RECOMMENDED CITATION

stats writer (2024). *How can the square root function be used in R and what are some examples of its implementation?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=141861>

The square root function in R is a mathematical function that calculates the square root of a given number. This function can be used in various ways in R programming, such as to find the length of a side of a square or to solve quadratic equations. It is a useful tool for data analysis and manipulation, as it can be applied to datasets to find the square root of each value. For example, the square root function can be used to analyze stock market data or to calculate the standard deviation of a dataset. Its implementation can also be seen in statistical tests, such as the t-test or ANOVA, where the square root function is used to calculate the standard error. Overall, the square root function is a versatile tool in R that can be applied to various mathematical and statistical tasks, making it an essential function for data scientists and analysts.

Use the Square Root Function in R (With Examples)

You can use the `sqrt()` function to find the square root of a numeric value in R:

`sqrt(x)`

The following examples show how to use this function in practice.

Example 1: Calculate Square Root of a Single Value

The following code shows how to calculate the square root of a single value in R:

```
#define x
```

```
x <- 25
```

```
#find square root of x
```

sqrt(x)

5

Example 2: Calculate Square Root of Values in Vector

The following code shows how to calculate the square root of every value in a vector in R:

```
#define vector
```

```
x <- c(1, 3, 4, 6, 9, 14, 16, 25)
```

```
#find square root of every value in vector
```

```
sqrt(x)
```

```
1.000000 1.732051 2.000000 2.449490 3.000000 3.741657  
4.000000 5.000000
```

Note that if there are negative values in the vector, there will be a warning message. To avoid this warning message, you can first convert each value in the vector to an absolute value:

```
#define vector with some negative values
```

```
x <- c(1, 3, 4, 6, -9, 14, -16, 25)
```

```
#attempt to find square root of each value in vector  
sqrt(x)
```

```
1.000000 1.732051 2.000000 2.449490 NaN 3.741657 NaN  
5.000000
```

Warning message:

In sqrt(x) : NaNs produced

```
#convert each value to absolute value and then find  
square root of each value  
sqrt(abs(x))
```

```
1.000000 1.732051 2.000000 2.449490 3.000000 3.741657  
4.000000 5.000000
```

Example 3: Calculate Square Root of Column in Data Frame

The following code shows how to calculate the square root of a single column in a data frame:

```
#create data frame  
data <- data.frame(a=c(1, 3, 4, 6, 8, 9),  
b=c(7, 8, 8, 7, 13, 16),  
c=c(11, 13, 13, 18, 19, 22),  
d=c(12, 16, 18, 22, 29, 38))
```

```
#find square root of values in column a  
sqrt(data$a)
```

```
1.000000 1.732051 2.000000 2.449490 2.828427 3.000000
```

Example 4: Calculate Square Root of Several Columns in Data Frame

The following code shows how to use the apply() function to calculate the square root of several columns in a data frame:

```
#create data frame  
data <- data.frame(a=c(1, 3, 4, 6, 8, 9),  
b=c(7, 8, 8, 7, 13, 16),  
c=c(11, 13, 13, 18, 19, 22),  
d=c(12, 16, 18, 22, 29, 38))
```

```
#find square root of values in columns a, b, and d  
apply(data, 2, sqrt)
```

```
a b d  
1.000000 2.645751 3.464102  
1.732051 2.828427 4.000000  
2.000000 2.828427 4.242641  
2.449490 2.645751 4.690416  
2.828427 3.605551 5.385165
```

3.000000 4.000000 6.164414

ARABPSYCHOLOGY.COM