

How can I create a nested for loop in R and what are some examples of its usage?

Authored by
stats writer

April 21, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I create a nested for loop in R and what are some examples of its usage?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137640>

A nested for loop in R is a programming structure that allows for the repetition of a specific task or set of tasks within another loop. It involves using one loop inside another loop, creating a "loop within a loop" structure. This can be achieved by placing one for loop inside the body of another for loop.

An example of using a nested for loop in R would be in creating a matrix or a two-dimensional array. The outer loop can be used to iterate through the rows, while the inner loop can be used to iterate through the columns, filling in the values for each element in the matrix.

Another example is when dealing with multiple data frames or lists. The outer loop can be used to iterate through each data frame or list, while the inner loop can be used to perform operations on each individual element within the data frame or list.

In general, nested for loops are useful in situations where a task needs to be repeated multiple times, with each repetition requiring a slightly different set of conditions. It allows for more complex and precise control over the execution of code and can greatly improve the efficiency of certain tasks. It is important to carefully plan and structure the nested loops to avoid unintended consequences and potential errors.

Create a Nested For Loop in R (Including Examples)

A nested for loop allows you to loop through elements in multiple vectors (or multiple dimensions of a matrix) and perform some operations.

The basic structure of a for loop in R is:

```
for(i in 1:4) {  
  print (i)  
}
```

1

2

3

4

And the basic structure of a nested for loop is:

```
for(i in 1:4) {  
  for(j in 1:2) {  
    print (i*j)  
  }  
}
```

1

2

2

4

3

6

4

8

This tutorial shows a few examples of how to create nested for loops in R.

Example 1: Nested For Loop in R

The following code shows how to use a nested for loop to fill in the values of a 4×4 matrix:

```
#create matrix
```

```
empty_mat <- matrix(nrow=4, ncol=4)
```

```
#view empty matrix
```

```
empty_mat
```

```
NA NA NA NA
```

```
NA NA NA NA
```

```
NA NA NA NA
```

```
NA NA NA NA
```

```
#use nested for loop to fill in values of matrix
```

```
for(i in 1:4) {
```

```
  for(j in 1:4) {
```

```
    empty_mat = (i*j)
```

```
  }
```

```
}
```

```
#view matrix
```

```
empty_mat
```

```
1 2 3 4
```

```
2 4 6 8
```

3 6 9 12

4 8 12 16

Example 2: Nested For Loop in R

The following code shows how to use a nested for loop to square each value in a data frame:

```
#create empty data frame
```

```
df <- data.frame(var1=c(1, 7, 4),  
var2=c(9, 13, 15))
```

```
#view empty data frame
```

```
df
```

```
var1 var2
```

```
1 1 9
```

```
2 7 13
```

```
3 4 15
```

```
#use nested for loop to square each value in the data  
frame
```

```
for(i in 1:nrow(df)) {
```

```
for(j in 1:ncol(df)) {
```

```
df = df^2
```

```
}
```

```
}
```

```
#view new data frame
```

```
df
```

```
var1 var2
```

```
1 1 81
```

```
2 49 169
```

```
3 16 225
```

A Note on Looping

In general, nested for loops perform fine on small datasets or matrices but they tend to be fairly slow with larger data.

For big data, the family of apply functions tend to be much quicker and the data.table package has many built-in functions that perform efficiently on larger datasets.

[How to Loop Through Column Names in R](#)

[How to Append Rows to a Data Frame in R](#)