How can we use do.call in R? (3 Examples)

Authored by

stats writer

June 29, 2024

ARABPSYCHOLOGY.COM

RECOMMENDED CITATION

stats writer (2024). *How can we use do.call in R? (3 Examples)*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=158824

Do.call is a function in R that allows for the execution of another function with a list of arguments. This is useful in situations where we want to apply a function to multiple variables or objects at once. By using do.call, we can save time and effort in writing repetitive code. This can be achieved by passing the function name and a list of arguments to the do.call function. For example, we can use do.call to apply a function to a list of data frames, merge multiple data frames, or even create new functions dynamically. Overall, do.call is a powerful tool in R that helps streamline coding processes and increase efficiency.

Use do.call in R (3 Examples)

You can use do.call() in R to apply a given function to a list as a whole.

This function uses the following basic syntax:

do.call(function, list)

The following examples show how to use do.call() in practice.

Example 1: Use do.call() with sum

The following code shows how to use do.call() to calculate the sum of values in a list:

```
#create list
values_list <- list(A=c(1, 2, 3), B=c(7, 5, 10), C=c(9, 9, 2))

#calculate sum of values in list
```

ARABPSYCHOLOGY.COM

```
do.call(sum, values_list)

48
```

The sum of the values in the list is 48.

Note that we would receive an error if we just tried to use sum() directly with the list:

```
#create list
values_list <- list(A=c(1, 2, 3), B=c(7, 5, 10), C=c(9, 9, 2))

#attempt to sum values in list
sum(values_list)

Error in sum(values_list) : invalid 'type' (list) of argument
```

Example 2: Use do.call() with mean

The following code shows how to use do.call() to calculate the mean of values in a list:

```
#define argument to use in do.call
args <- list(1:20, na.rm=TRUE)
```

ARABPSYCHOLOGY.COM

```
#calculate mean of values in list
do.call(mean, args)

10.5
```

The mean of the values in the list is 10.5.

Note that we would receive an error if we just tried to use mean() directly with the list:

```
#attempt to calculate mean of values in list
mean(list(1:20), na.rm=TRUE)

NA
Warning message:
In mean.default(list(1:20), na.rm = TRUE) :
argument is not numeric or logical: returning NA
```

Example 3: Use do.call() with rbind

```
#create three data frames
df1 <- data.frame(team=c('A', 'B', 'C'),
points=c(22, 27, 38))

df2 <- data.frame(team=c('D', 'E', 'F'),
```

ARABPSYCHOLOGY.COM

```
points=c(22, 14, 20))

df3 <- data.frame(team=c('G', 'H', 'I'),
points=c(11, 15, 18))

#place three data frames into list
df_list <- list(df1, df2, df3)

#row bind together all three data frames
do.call(rbind, df_list)

team points
1 A 22
2 B 27
3 C 38
4 D 22
5 E 14
6 F 20
7 G 11
8 H 15
9 I 18
```

The result is one data frame that contains the rows from each of the three data frames.

ARABPSYCHOLOGY.COM

Note that we would not receive the desired data frame if we tried to use rbind() directly with the list:

```
#create three data frames
df1 <- data.frame(team=c('A', 'B', 'C'),
points=c(22, 27, 38))

df2 <- data.frame(team=c('D', 'E', 'F'),
points=c(22, 14, 20))

df3 <- data.frame(team=c('G', 'H', 'I'),
points=c(11, 15, 18))

#place three data frames into list
df_list <- list(df1, df2, df3)

#attmempt to row bind together all three data frames
rbind(df_list)

df_list List,2 List,2 List,2
```

ARABPSYCHOLOGY.COM

Additional Resources

How to Use paste & paste0 Functions in R