

How can the source function be used in R? Can you provide an example?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *How can the source function be used in R? Can you provide an example?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=159636>

The source function in R is used to load and execute an external R script or file. This allows users to easily import and run code from other sources in their current R session. To use the source function, the file or script must be located in the current working directory or the full path to the file must be specified. An example of using the source function is as follows: if we have a file named "example.R" in our current working directory, we can use the command "source("example.R")" to load and execute the code in that file. This can be particularly useful for running multiple scripts or functions from a single R session, as well as for sharing and collaborating on code with others.

Use the source Function in R (With Example)

You can use the source function in R to reuse functions that you create in another R script.

This function uses the following basic syntax:

```
source("path/to/some/file.R")
```

Simply add this line to the top of your R script and you'll be able to use any functions defined in file.R.

The following example shows how to use the source function in practice.

Example: Using the source Function in R

Suppose we have the following R script called `some_functions.R` that contains two simple user-defined functions:

#define function that divides values by 2

```
divide_by_two <- function(x) {  
  return(x/2)  
}
```

#define function that multiplies values by 3

```
multiply_by_three <- function(x) {  
  return(x*3)  
}
```

Now suppose we're currently working with some R script called main_script.R.

Assuming some_functions.R and main_script.R are located within the same folder, we can use source at the top of our main_script.R to allow us to use the functions we defined in the some_functions.R script:

```
source("some_functions.R")
```

#create data frame

```
df <- data.frame(team=c('A', 'B', 'C', 'D', 'E', 'F'),  
  points=c(14, 19, 22, 15, 30, 40))
```

#view data frame

df

team points

1 A 14

2 B 19

3 C 22

4 D 15

5 E 30

6 F 40

**#create new columns using functions from
some_functions.R**

df\$half_points <- divide_by_two(df\$points)

df\$triple_points <- multiply_by_three(df\$points)

#view updated data frame

df

team points half_points triple_points

1 A 14 7.0 42

2 B 19 9.5 57

3 C 22 11.0 66

4 D 15 7.5 45

5 E 30 15.0 90

6 F 40 20.0 120

Notice that we're able to create two new columns in our data frame using functions that we defined in the `some_functions.R` script.

The source function allowed us to use the `divide_by_two` and `multiply_by_three` functions in our current script, even though these functions weren't created in the current script.

Note: In this example, we only used one source function at the top of the file. However, we can use as many source functions as we'd like if we want to reuse functions defined in several different scripts.

Additional Resources