What is the difference between require() and library() in R?

Authored by

stats writer

June 25, 2024

ARABPSYCHOLOGY.COM

RECOMMENDED CITATION

stats writer (2024). *What is the difference between require() and library() in R?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=152805

Require() and library() are two functions in the R programming language that are used to load external packages or libraries into the current R session. The main difference between these two functions is their behavior when the specified package is not available.

Require() will check if the package is already installed on the system and if it is not, an error will be thrown. On the other hand, library() will attempt to install the package if it is not already available. This makes library() a more user-friendly option as it will automatically install any missing packages, whereas require() requires the user to manually install the package before using it.

Another difference is that require() is a base R function, while library() is part of the recommended package "utils". This means that require() can be used without loading any additional packages, whereas library() requires the "utils" package to be loaded first.

In summary, require() and library() serve similar purposes in loading external packages, but differ in their behavior when the package is not available.

The Difference Between require() and library() in R

The require() and library() functions can both be used to load packages in R, but they have one subtle difference:

require() will output a warning if a package is not installed and then continue to execute the code.library() will output an error and stop the execution of the code.

Because of this difference, require() is usually only used if you are loading packages inside a function so that the function will continue to execute even if a package doesn't exist.

In practice, most programmers recommend using

ARABPSYCHOLOGY.COM

library() since you'll want to receive an error message that lets you know a package is not installed.

This is something you want to be aware of as early on as possible when writing code.

The following example illustrates the difference between the require() and library() functions in practice.

Example: The Difference Between require() and library() in R

Suppose we would like to load the BostonHousing dataset from the mlbench package, but assume that we don't have the mlbench package already installed.

The following code shows how to use the library() function to attempt to load this package and perform some data analysis on the BostonHousing dataset:

```
#attempt to load mlbench library
library(mlbench)

Error in library(mlbench) : there is no package called 'mlbench'
#load Boston Housing dataset
data(BostonHousing)
```

ARABPSYCHOLOGY.COM

#view summary of Boston Housing dataset
summary(BostonHousing)

#view total number of rows in Boston Housing dataset
nrow(BostonHousing)

Since the mlbench package is not already installed, we receive an error when we use the library() function and the rest of the code is not even executed.

This is helpful since it immediately makes us aware that this package is not installed and that we should install it before proceeding.

However, suppose we instead used require() to load the mlbench package:

#attempt to load mlbench library
require(mlbench)

Warning message:
In library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE, :
there is no package called 'mlbench'
#load Boston Housing dataset

ARABPSYCHOLOGY.COM

data(BostonHousing)

Warning message:
In data(BostonHousing) : data set 'BostonHousing' not found
#view summary of Boston Housing dataset
summary(BostonHousing)

Error in summary(BostonHousing) : object 'BostonHousing' not found
#view total number of rows in Boston Housing dataset
nrow(BostonHousing)

In this example, we don't receive an error message until we try to use the summary() function to summarize the BostonHousing dataset.

Instead, we receive a warning after using the require() function and the rest of the code continues to run until we encounter an error.

This example illustrates the difference between library() and require() in R: The library() function produces an error immediately and doesn't execute the rest of the code since mlbench is not loaded.

ARABPSYCHOLOGY.COM

Bonus: Check if Particular Package is Installed

We can use the system.file() function to check if a particular package is installed in our current R environment.

For example, we can use the following syntax to check if the ggplot2 package is installed in the current R environment:

```
#check if ggplot2 is installed
system.file(package='ggplot2')
```

```
"C:/Users/bob/Documents/R/win-library/4.0/ggplot2"
```

Since ggplot2 is installed, the function simply returns the file path where the package is installed.

Now suppose we check if the mlbench package is installed:

```
#check if mlbench is installed
system.file(package='mlbench')
```

```
""
```

ARABPSYCHOLOGY.COM

The function returns an empty string, which tells us that the mlbench package is not installed in our current environment.

ARABPSYCHOLOGY.COM