

How can I import CSV files into R using a step-by-step process?

Authored by
stats writer

April 21, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I import CSV files into R using a step-by-step process?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137733>

Importing CSV files into R can be done easily by following a simple step-by-step process. First, open R and make sure the "readr" package is installed by typing "install.packages("readr")" in the console. Then, set your working directory to the location where the CSV file is saved using the "setwd()" command. Next, use the "read_csv()" function from the readr package to read in the CSV file and assign it to a variable. Finally, use the "head()" function to view the first few rows of the imported file and ensure it was imported correctly. This process allows for smooth and efficient importing of CSV files into R for data analysis and manipulation.

Import CSV Files into R (Step-by-Step)

Suppose I have a CSV file called data.csv saved in the following location:

C:\Users\Bob\Desktop\data.csv

And suppose the CSV file contains the following data:

team, points, assists

'A', 78, 12

'B', 85, 20

'C', 93, 23

'D', 90, 8

'E', 91, 14

There are three common ways to import this CSV file into R:

1. Use read.csv from base R (Slowest method, but

works fine for smaller datasets)

```
data1 <- read.csv("C:UsersBobDesktopdata.csv",  
header=TRUE, stringsAsFactors=FALSE)
```

2. Use read_csv from readr package (2-3x faster than read.csv)

```
library(readr)
```

```
data2 <- read_csv("C:UsersBobDesktopdata.csv")
```

3. Use fread from data.table package (2-3x faster than read_csv)

```
library(data.table)
```

```
data3 <- fread("C:UsersBobDesktopdata.csv")
```

This tutorial shows an example of how to use each of these methods to import the CSV file into R.

Method 1: Using read.csv

If your CSV file is reasonably small, you can just use the read.csv function from Base R to import it.

When using this method, be sure to specify `stringsAsFactors=FALSE` so that R doesn't convert character or categorical variables into factors.

The following code shows how to use `read.csv` to import this CSV file into R:

```
#import data  
data1 <- read.csv("C:UsersBobDesktopdata.csv",  
header=TRUE, stringsAsFactors=FALSE)
```

```
#view structure of data  
str(data1)
```

```
'data.frame': 5 obs. of 3 variables:  
$ team : chr "A" "B" "C" "D" ...  
$ points : int 78 85 93 90 91  
$ assists: int 12 20 23 8 14
```

Method 2: Using `read_csv`

If you're working with larger files, you can use the `read_csv` function from the `readr` package:

```
library(readr)
```

```
#import data
```

```
data2 <- read_csv("C:UsersBobDesktopdata.csv")
```

```
#view structure of data
```

```
str(data2)
```

```
'data.frame': 5 obs. of 3 variables:
```

```
$ team : chr "A" "B" "C" "D" ...
```

```
$ points : int 78 85 93 90 91
```

```
$ assists: int 12 20 23 8 14
```

Method 3: Using fread

If your CSV is extremely large, the fastest way to import it into R is with the fread function from the data.table package:

```
library(data.table)
```

```
#import data
```

```
data3 <- fread("C:UsersBobDesktopdata.csv")
```

```
#view structure of data
```

```
str(data3)
```

Classes 'data.table' and 'data.frame': 5 obs. of 3

variables:

```
$ team : chr "A" "B" "C" "D" ...
```

```
$ points : int 78 85 93 90 91
```

```
$ assists: int 12 20 23 8 14
```

Note that in each example we used double backslashes () in the file path to avoid the following common error:

Error: 'U' used without hex digits in character string starting ""C:U"

The following tutorials explain how to import other file types into R: