How can I use read.table in R?

Authored by

stats writer

July 1, 2024

Read.table is a function in the R programming language that allows users to read and import tabular data into their R environment. This function is commonly used for reading data from CSV, TXT, or other delimited files. It offers various options for customizing the import process, such as specifying the file path, choosing the appropriate data types, and handling missing values. By using read.table, users can efficiently load and manipulate large datasets in R for further analysis and visualization.

Use read.table in R (With Examples)

You can use the read.table function to read in a file that contains tabular data into R.

This function uses the following basic syntax:

df <- read.table(file='C:UsersbobDesktopdata.txt', header=FALSE, sep = "")

By default, the read.table function assumes there is no header row in the file and that the values are separated by white space.

However, you can use the header and sep arguments to tell R that the file has a header row and uses a different delimiter.

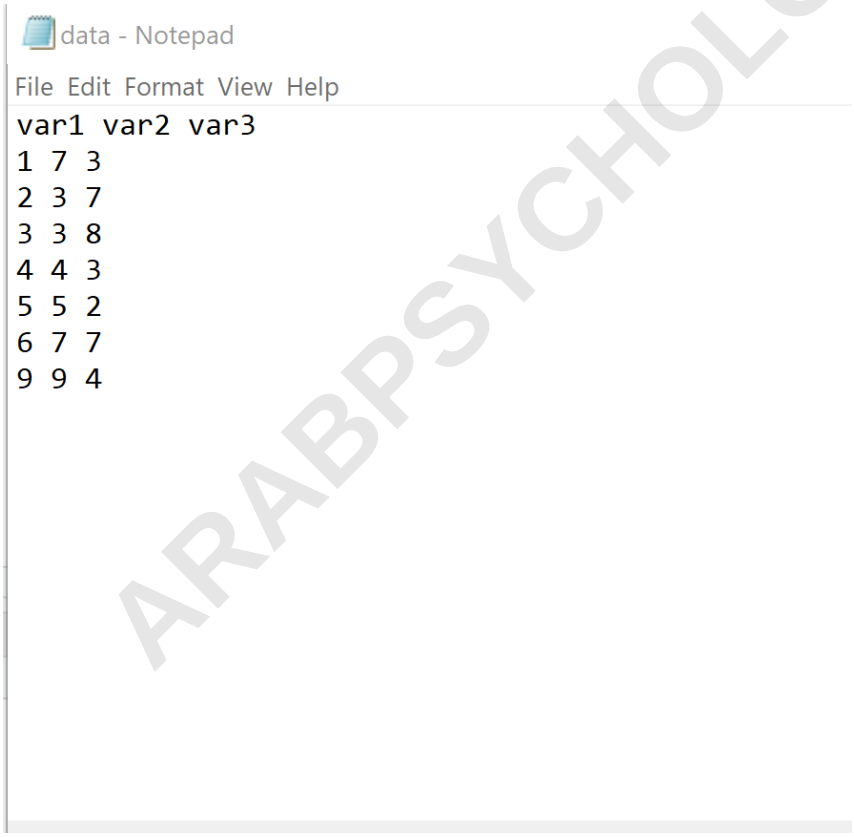
For example, you could choose to use a comma as a delimiter:

df <- read.table(file='C:UsersbobDesktopdata.txt', header=TRUE, sep=',')

The following step-by-step example shows how to use the read.table function in practice.

Step 1: View the File

Suppose I have a file called data.txt on my Desktop that I'd like to read into R as a data frame:



Step 2: Use read.table() to Read File into Data Frame

Next, let's use read.table() to read the file into a data frame called df:

#read file from Desktop into data frame
df <- read.table(file='C:UsersbobDesktopdata.txt', header=TRUE)

Note that I specified header=TRUE since the first row in the file contains column names.

I also didn't specify the sep argument since the data in the file is already separated by white space.

Step 3: View the Data Frame

#view data frameprint(df)

var1 var2 var3
1 1 7 3
2 2 3 7
3 3 3 8
4 4 4 3
5 5 5 2
6 6 7 7
7 9 9 4

We can see that the data frame matches the data in the file.

We can also use the class and dim functions to check the class of the data frame and get the dimensions (number of rows and number of columns):

```
#check class of data frame
class(df)

"data.frame"

#check dimensions of data frame
dim(df)

7 3
```

We can see that df is indeed a data frame and has 7 rows and 3 columns.

Additional Resources

The following tutorials explain how to read other types of files into R: