

How can I convert a list into a dataframe in R?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I convert a list into a dataframe in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=150091>

To convert a list into a dataframe in R, the function "as.data.frame()" can be used. This function takes a list as its input and returns a dataframe with the same elements as the list. The dataframe will have column names corresponding to the names of the list elements, and each list element will be a column in the dataframe. This allows for efficient organization and analysis of data in R, as dataframes are a commonly used data structure. Additionally, various functions and packages in R are specifically designed to work with dataframes, making them a useful tool for data manipulation and analysis.

Let's see how to convert a list and nested list to R DataFrame in other words create a data frame in R from list. A list is a one-dimensional data structure that can store multiple data-type elements. In this article, I will explain how to convert a list and nested list to an R DataFrame.

You can create a list in R by using the `list()` function. Use `data.frame()` method to convert any data structure to the list. And also if we want to convert nested lists to data frames, we can use `do.call()` functions column and row-wise. `as.data.frame()` is also used to convert nested lists to a dataframe.

Steps to Covert List to DataFrame

1. Quick Examples of Convert List to R DataFrame

If you are in a hurry, below are quick examples of how to convert a list to DataFrame or create a DataFrame from the list.

```
# Below are the quick examples of converting list to dataframe
# Create list with 5 elements
my_input_list = list(56,78,90,45,67)
```

```
# Example 1: Convert dataframe to list using data.frame()
df <- data.frame(my_input_list)
```

```
# Example 2: Convert dataframe to list using data.frame()
df <- data.frame(t(sapply(my_input_list,c)))
```

```
# Example 3: Create nested list (3 lists inside)
my_nested_list = list(id=list(1,2,3,4,5),
name=list('sai','ram','hari','deepak','sahithi'),
gender=list('m','m','m','m','f'))
```

```
# Convert nested list to the dataframe by columns
df <- as.data.frame(do.call(cbind, my_nested_list))
```

```
# Example 4: Convert nested list to the dataframe by rows
```

```
df <- as.data.frame(do.call(rbind, my_nested_list))
```

```
# Example 5: Using data.table
```

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

```
df <- rbindlist(my_nested_list)
```

```
# Example 6: Using tibble
```

```
library(tibble)
```

```
df <- enframe(my_nested_list)
```

```
# Example 7: Using dplyr package
```

```
library(dplyr)
```

```
df <- as_tibble(my_nested_list)
```

What is DataFrame in R?

A **data frame** in R is a fundamental data structure that is used for storing and manipulating structured data which is in the format of rows and columns similar to an RDBMS table or spreadsheet. It is a two-dimensional data structure such that one dimension refers to the row and another dimension refers to a column. Each column in the data frame is a Vector of the same length, in other words, all columns in the data frame should have the same length.

Data frame columns are referred to as variables and rows are referred to as observations. If you are new to R Programming, I would highly recommend reading the [R Programming Tutorial](#) where I have explained R concepts with examples.

What is List in R

In R, a list is a versatile and flexible data structure that can hold elements of different data types, including vectors, matrices, data frames, and even other lists. It is essentially an ordered collection of objects, and each element in the list can be accessed using an index. Lists are useful when you want to store and organize heterogeneous data in a single data structure. They provide a way to represent complex structures and are widely used in R programming.

To convert a list to R DataFrame, first, create a list in R using the `list()` function. This function takes the comma-separated values and returns a list type.

Syntax:

```
# List creation syntax  
list(elements)
```

Where `elements` can be a `vector`, `array`, `matrix`, `dataframe` or even another list.

Let's create a list with 5 elements and display them.

```
# Create list with 5 elements  
my_input_list = list(56,78,90,45,67)  
  
# Display the list  
print(my_input_list)  
  
# Type of the object  
class(my_input_list)
```

Output:

```
> # Display the list
[[1]]
[1] 56

[[2]]
[1] 78

[[3]]
[1] 90

[[4]]
[1] 45

[[5]]
[1] 67

> # Type of the object
> class(my_input_list)
[1] "list"
```

Now, let's see how to convert the list to an R DataFrame.

2. Convert List to DataFrame using data.frame()

`data.frame()` is used to create a DataFrame in R that takes a list, vector, array, etc as arguments, Hence, we can pass a given list to the `data.frame()` function to convert the list to DataFrame. It will store the elements in a single row in the DataFrame. In the converted DataFrame, the column names will be X followed by the element from the list.

Syntax:

```
# Syntax create DataaFrame from list
data.frame(my_input_list)
```

Let's convert the above-created list to the DataFrame.

```
# Convert dataframe to list using data.frame()
df <- data.frame(my_input_list)
df
```

Yields below output.

```
   X56 X78 X90 X45 X67
1   56  78  90  45  67
```

Notice that the DataFrame is created from the list. This DataFrame contains 1 row with the number of columns equal to the number of elements in an array.

3. Convert List to a DataFrame using t()

`t()` stands for transpose. We can transpose a list to a DataFrame such that `t()` will take a parameter `sapply()`.

`sapply()` take two parameters. The first parameter is the list name and the second is the `c`, which stands for the combine. Again, in the converted DataFrame, the column names will be X characters followed by the element count.

Syntax:

```
# Syntax
data.frame(t(sapply(my_input_list,c)))
```

Let's convert the above-created list to the DataFrame.

```
# Convert dataframe to list using data.frame()
df <- data.frame(t(sapply(my_input_list,c)))
df
```

Yields below output.

```
# Output
x1 x2 x3 x4 x5
1 56 78 90 45 67
```

4. Convert Nested List to a DataFrame

`as.data.frame()` converts the nested list to an R DataFrame by taking `do.call()` function as a parameter. So each list inside a nested list will be a column in a DataFrame. So the column names in the DataFrame will be nested list names.

Syntax:

```
# Syntax using as.data.frame()
as.data.frame(do.call(cbind, my_nested_list))
```

Where the `do.call()` takes two parameters.

Parameters:

`cbind` stands for column binding that places elements column-wise in the converted DataFrame. `my_nested_list` is the input nested list.

Example:

Let's create a nested list (3 lists inside) and convert it to the DataFrame.

```
#Create nested list (3 lists inside)
my_nested_list <- list(id=list(1,2,3,4,5),
name=list('sai','ram','hari','deepak','sahithi'),
gender=list('m','m','m','m','f'))

# Convert nested list to the dataframe by columns
df <- as.data.frame(do.call(cbind, my_nested_list))
df
```

Yields below output.

```
# Output
id name gender
1 1 sai m
2 2 ram m
3 3 hari m
4 4 deepak m
5 5 sahithi f
```

We can see that the nested list is converted to a DataFrame as column values.

5. Convert Nested List to a DataFrame by Rows

`as.data.frame()` converts the nested list to a DataFrame by taking `do.call()` function as a parameter. So each list inside a nested list will be a row in a DataFrame. So the row names in the DataFrame will be nested list names. More examples of creating a Data Frame refer to different ways to [create a Data Frame in R](#).

Syntax:

```
# Syntax
as.data.frame(do.call(rbind, my_nested_list))
```

Where `do.call()` takes two parameters.

Parameters:

`rbind` stands for row binding that places elements row-wise in the converted DataFrame. `my_nested_list` is the input nested list.

Example:

Let's create a nested list (3 lists inside) and convert it to the R DataFrame.

```
#Create nested list (3 lists inside)
my_nested_list = list(id=list(1,2,3,4,5),
name=list('sai','ram','hari','deepak','sahithi'),
gender=list('m','m','m','m','f'))

# Convert nested list to the dataframe by rows
df <- as.data.frame(do.call(rbind, my_nested_list))
```

Yields below output.

```
# Output
v1 v2 v3 v4 v5
id 1 2 3 4 5
name sai ram hari deepak sahithi
gender m m m m f
```

We can see that the nested list is converted to a DataFrame as row values.

6. Use data.table to Create DataFrame from List

The `package data.table` has the function `rbindlist` which is a superfast implementation. This returns a `data.table` and to convert to `data.frame` use `as.data.frame(DT)`. `rbindlist()` function. This also takes list of `lists`, `data.frames` or `data.tables` as input.

```
# Using data.table
library(data.table)
df <- rbindlist(my_nested_list)
df
```

Yields below output.

```
# Output
v1 v2 v3 v4 v5
1: 1 2 3 4 5
2: sai ram hari deepak sahithi
3: m m m m f
```

7. Using tibble Package

`tibble` is another package you can use to create a R DataFrame from list. The `tibble` package has a function `enframe()` which takes nested `list` objects and convert it to nested `tibble` ("tidy" data frame) objects.

```
# Using tibble
library(tibble)
```

```
df <- enframe(my_nested_list)
df
```

Yields below output.

```
# A tibble: 3 × 2
  name value
<chr> <list>
1 id <list >
2 name <list >
3 gender <list >
```

8. Create DataFrame from List in R using dplyr

In the following example, create a DataFrame from the list using the [R dplyr package](#). `dplyr` is a third-party package hence, you need to load the library using `library("dplyr")` to use its methods. In case you don't have this package, install it using `install.packages("dplyr")`.

For bigger data sets it is best to use the methods from `dplyr` package as they perform 30% faster. `dplyr` package uses C++ code to evaluate.

```
# Using dplyr package
library(dplyr)
df <- as_tibble(my_nested_list)
df
```

9. Use plyr Package to create DataFrame from List

Use `ldply()` from `plyr` package to create and returns a `data.frame` from list. All the data types (character, numeric, etc) are correctly transformed. If the list has different data types they will be all transformed to character with `matrix` approach.

```
# Using plyr package
library (plyr)
df <- ldply(my_nested_list, data.frame)
df
```

7. Conclusion

From this article, you have learned `data.frame()` and `as.data.frame()` can be used to convert a list to R DataFrame or create a data frame from a list. If you want the elements in the list column-wise, then use `cbind` otherwise you can use `rbind`. Also learned to use different third-party packages to create DataFrame from a list.

Related Articles

References

[R dataframe](#) [R List](#)

ARABPSYCHOLOGY.COM