

How can I create a DataFrame from vectors in R?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I create a DataFrame from vectors in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=150107>

Creating a DataFrame from vectors in R involves using the function 'data.frame' to combine multiple vectors into a single data structure. This function takes in the vectors as arguments and organizes them into columns, creating a table-like structure. This allows for easy manipulation and analysis of the data within the DataFrame. Additionally, the 'names' argument can be used to assign labels to the columns. Overall, using the 'data.frame' function is a simple and efficient way to create a DataFrame from vectors in R.

To create a data frame in R either from Vectors or by transforming a vector into a data frame using the `data.frame()` function. In R, vectors hold elements of uniform type, which can include logical, integer, double, character, complex, or raw types. Vectors are built using the `c()` function. Whereas the R data frame is a 2-dimensional structure that is used to hold the values in rows and columns. In the data frame, each column stores the values of one variable and each row stores the value of each column.

Create an R Vector

In R, a vector is a basic data structure that is used to store elements of the same data type and the types can be logical, integer, double, character, complex, or raw.

R Vector can be created by using `c()`. Let's see the syntax and create a vector.

```
# Syntax of c() function  
c(...)
```

Now let's create a Vector.

```
# Create Vectors  
id <- c(10,11,12,13)  
name <- c('sai','ram','deepika','sahithi')  
dob <- as.Date(c('1990-10-02','1981-3-24','1987-6-14','1985-8-16'))
```

Here are the different types of vector variables:

Let's display the type of these Vector variables.

```
# Types of Vectors  
> typeof(id)  
# "double"
```

```
> typeof(name)
# "character"
> typeof(dob)
# "double"
```

Create an R DataFrame from Vectors

Using the `data.frame()` function, we can create a DataFrame from vectors. A data frame is essentially a list of variables that share the same number of rows, each with unique row names. Therefore, all vectors used to construct a DataFrame must be of the same length; otherwise, an error will occur.

For syntax and usage of `data.frame()`, refer to [How to Create a data frame in R](#).

You need to follow the below guidelines when creating a DataFrame from Vector in R using `data.frame()` function.

Create DataFrame From Vectors Example

Now, let's create a data frame from Vectors by using `data.frame()` function. This function takes the first argument either list or vector. I will use the Vectors that are created above and pass them as arguments to `data.frame()` to create a data frame from Vectors.

```
# Create DataFrame
df <- data.frame(id,name,dob)

# Print DataFrame
df
```

In the above example, I have used the vectors `id`, `name` and `dob` as arguments to the `data.frame()` function, separated by commas. The above example yields the below output. The example above produces the following output. R generates a data frame, assigning column names/variables identical to those used for the vector. You can display the data frame on the console using `df` or `print(df)`.

```
# Output
id name dob
1 10 sai 1990-10-02
2 11 ram 1981-03-24
```

```
3 12 deepika 1987-06-14
4 13 sahithi 1985-08-16
```

You will observe that it automatically assigns an incremental sequence number to every row in a data frame.

Alternatively, you can generate a data frame by directly supplying the vector variables to the function.

```
# Create DataFrame
df <- data.frame(
  id = c(10,11,12,13),
  name = c('sai','ram','deepika','sahithi'),
  dob = as.Date(c('1990-10-02','1981-3-24','1987-6-14','1985-8-16'))
)

# Print DataFrame
df
```

Let's examine the data frame's types with `print(sapply(df, class))`. It's important to mention that I haven't explicitly set the data types for any column, so R will automatically determine them based on the data.

```
# Display datatypes
print(sapply(df, class))

# Output
# id name dob
# "numeric" "Factor" "Date"
```

Another way to utilize the `str(df)` is to examine the data types.

```
# Display datatypes
str(df)

# Output
# 'data.frame': 4 obs. of 3 variables:
# $ id : num 10 11 12 13
# $ name: Factor w/ 4 levels "deepika","ram",...: 4 2 1 3
```

```
# $ dob : Date, format: "1990-10-02" "1981-03-24" "1987-06-14" "1985-08-16"
```

Suppose you notice that the column comprises characters but is classified as a Factor data type. In that case, it's because R defaults to converting character vectors into factors when constructing a data frame with character vectors.

You can modify this behavior by including the extra parameter `stringsAsFactors=False` during data frame construction.

```
# Create DataFrame
df <- data.frame(
  id = c(10,11,12,13),
  name = c('sai','ram','deepika','sahithi'),
  dob = as.Date(c('1990-10-02','1981-3-24','1987-6-14','1985-8-16')),
  stringsAsFactors=FALSE
)
```

```
# Print DataFrame
```

```
str(df)
```

Produces the following output. It's important to note that the data type for the `name` column/variable is a character (`chr`). In R, it's frequently necessary to convert the data frame from `Factor` to `Character` before conducting certain operations or transformations.

```
# Output
'data.frame': 4 obs. of 3 variables:
 $ id : num 10 11 12 13
 $ name: chr "sai" "ram" "deepika" "sahithi"
 $ dob : Date, format: "1990-10-02" "1981-03-24" "1987-06-14" ...
```

Complete Example

Following is a complete example of how to create a data frame from a Vectors in R or how to convert a Vector into a data frame.

```
# Create R Vectors
id <- c(10,11,12,13)
name <- c('sai','ram','deepika','sahithi')
```

```
dob <- as.Date(c('1990-10-02', '1981-3-24', '1987-6-14', '1985-8-16'))

# Check Vector types
typeof(id)
typeof(name)
typeof(dob)

# Create DataFrame
df <- data.frame(id,name,dob)
df

# Display DataFrame datatypes
print(sapply(df, class))
```

Conclusion

In this article, you have learned how to create a Vector and convert Vectors to an R data frame using examples. The vector consists of elements of uniform type, which can include logical, integer, double, character, complex, or raw data. You can construct a vector using the `c()` function. On the other hand, an R data frame represents a two-dimensional structure designed to store values in rows and columns.

You can find several R examples at [Github R Programming Examples Project](#).

Related Articles

References