

How can I add a row to a dataframe in R?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I add a row to a dataframe in R?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=149580>

Adding a row to a dataframe in R is a simple process that involves using the "rbind()" function. This function allows you to combine a new row of data with an existing dataframe. To add a row, you first need to create a vector or list containing the values for each column in the new row. Then, you can use the "rbind()" function to combine this new row with the existing dataframe, and assign the result to a new variable. Alternatively, you can also use the "add_row()" function from the "dplyr" package to add a row to a dataframe. Both methods allow you to easily expand and modify your dataframe in R.

To add a new row(or rows) to DataFrame in R, you can use various functions. Adding is one of the data manipulation tasks of R programming. We know that the data frame in R language is a major data structure that can hold the data of different data types. So, we can easily modify these data using various approaches of R. Adding a new observation(row) to the data frame is one of the typical modifications in R.

In this article, I will explain how easy to add single/multiple rows to the DataFrame. The following methods are used to add a row to DataFrame in R.

[rbind\(\) from R base](#) [add_row\(\) from tibble or tidyverse](#) [rows_insert\(\) from dplyr](#)

1. Quick Examples of Adding Row to DataFrame

The following are quick examples of how to add rows to a data.frame in R.

```
# Quick Examples of adding row to data frame
```

```
# Example 1: Add Row to the DataFrame
```

```
# This converts all types to string
```

```
df <- c(33, 50, "java")
```

```
# Example 2: Add Row to the DataFrame
```

```
# with out changing data types
```

```
df <- list(33, 50, "java")
```

```
# Example 3: Add Row using rbind()
```

```
new_row = c(id = 33, pages=50, name = "java")
```

```
df3 = rbind(df,new_row)
```

```
# Example 4: Add data.frame with another
```

```
df2 = data.frame(id = 33, pages=50, name = "java")
```

```
df3 = rbind(df,df2)
```

```
# Example 5: using tibble/tidyverse
library(tidyverse)
df2 <- df %>%
add_row(id = 33, pages=50, name = "java")

# Example 6: Using dplyr
library(dplyr)
df2 = data.frame(id = 33, pages=50, name = "java")
df3 <- df %>%
rows_insert(df2)
```

Let's create an R DataFrame, run these examples, and explore the output. If you already have data in CSV you can easily import CSV files to R DataFrame. Also, refer to Import Excel File into R.

```
# Create dataframe
df = data.frame(id=c(11,22),
pages=c(32,45),
name=c("spark", "python"))
df
```

Yields below output.

```
id pages  name
11   32  spark
22   45  python
```

2. Add Row to DataFrame

To add a new row to the DataFrame (`data.frame`) in R, first, you need to get the number of rows in the existing data frame used `nrows(df)` and add a new row towards the end `nrow(df) + 1`. The following example adds a new row at the end of the dataframe.

```
# Add Row to the DataFrame
# This converts all types to string
df <- c(33, 50, "java")
df
```

```
# Output
# id pages name
# 1 11 32 spark
# 2 22 45 python
# 3 33 50 java
```

When you have mixed data types some columns are numeric and some columns are text, then it converts all columns to text. You can check this by running `str(df)`.

Get the type of given data frame columns.

```
'data.frame':  3 obs. of  3 variables:
 $ id   : chr  "11" "22" "33"
 $ pages: chr  "32" "45" "50"
 $ name : chr  "spark" "python" "java"
```

3. Add Rows by Not Changing Data Types

You can use `list()` function instead of a vector to add a new row. In this example, specify the new row values using `list()` function. In this case it doesn't change the types of columns. You can run `str(df)` and validate the result yourself.

```
# Add Row to the DataFrame
# with out changing data types
df <- list(33, 50, "java")
df
```

```
# Output:
# id pages name
# 1 11 32 spark
# 2 22 45 python
# 3 33 50 java
```

```
# Get the type of the columns
str(df)
```

```
# Output:
# 'data.frame': 3 obs. of 3 variables:
# $ id : num 11 22 33
```

```
# $ pages: num 32 45 50  
# $ name : chr "spark" "python" "java"
```

4. Add Row using rbind()

The `rbind()` function in R, short for *row-bind*, is used to combine vectors, lists, matrices, and data frames by rows. The following example adds a list as a row to the DataFrame.

```
# Add Row using rbind()  
new_row = list(id = 33, pages=50, name = "java")  
df3 = rbind(df,new_row)  
df3
```

Yields the same as output the above.

5. Add DataFrame as a Row

You can also use `rbind()` function to append the new row by dataframe. When you do this, you have to set consistent column names for all the data frames you want to append.

```
# Add data.frame with another  
df2 = data.frame(id = 33, pages=50, name = "java")  
df3 = rbind(df,df2)  
df3
```

Yields the same as output the above.

6. Use add_row() from tibble or tidyverse

The packages `tibble` or `tidyverse` provide a function `add_row()` to add a row to DataFrame in R. This is a convenient way to add one or more rows of data to an existing data frame.

To use this function first, you need to install R package by using `install.packages("tidyverse")` and load it using the `library("tidyverse")`.

```
# syntax  
add_row(.data, ..., .before = NULL, .after = NULL)
```

Example

```
# using add_row() From tibble or tidyverse
library(tidyverse)
df2 <- df %>%
add_row(id = 33, pages=50, name = "java")
df2
```

7. Use rows_insert() from dplyr

Finally, use `rows_insert()` function from the [dplyr package](#) to add a row to R DataFrame. You need to install and load this package to use this function.

```
# Using rows_insert() from dplyr
library(dplyr)
df2 = data.frame(id = 33, pages=50, name = "java")
df3 <- df %>%
rows_insert(df2)
df3
```

Frequently Asked Questions on Add Row to DataFrame in R

How do I add a row to a DataFrame in R?

You can use the `rbind()` function to add a row to a DataFrame. For example, `new_row <- c(1, "Java", 25)`

```
df <- rbind(df, new_row).
```

How can I add a row with missing values?

You can add a row with missing values. If your DataFrame has columns with different data types, R will try to coerce the values to a common type, and missing values (`NA`) can be used for this purpose.

What if the data types don't match the DataFrame's structure?

If the data types of the new row don't match the existing DataFrame, R will attempt to coerce them. If coercion is not possible, you may encounter errors or unexpected behavior. It's generally good practice to ensure that the data types match or can be coerced appropriately.

Are there other methods to add rows to a DataFrame?

Besides `rbind()`, you can also use the `bind_rows()` function from the `dplyr` package:

How can I add multiple rows at once?

you can add multiple rows using the same `rbind()` or `bind_rows()` approach. Just make sure that the structure of the new rows matches the DataFrame.

8. Conclusion

In this article, you have learned how to add a new row(or rows) to DataFraem in R by using the `rbin()` function base library, `add_row()` from tibble or tidyverse, and `rows_inser()` from dplyr.

Related Articles

References