Joining of Dataframes in R Programming

Difficulty Level : Basic • Last Updated : 23 May, 2022

rreIn R Language, dataframes are generic data objects which are used to store the tabular data. Dataframes are considered to be the most popular data objects in R programming because it is more comfortable to analyze the data in the tabular form. Dataframes can also be taught as mattresses where each column of a matrix can be of the different data types. Dataframe is made up of three principal components, the data, rows, and columns. In R we use **merge()** function to merge two dataframes in R. This function is present inside **join()** function of **dplyr** package. The most important condition for joining two dataframes is that the column type should be the same on which the merging happens. **merge()** function works similarly like join in DBMS. Types of Merging Available in R are,

1. Natural Join or Inner Join
2. Left Outer Join
3. Right Outer Join
4. Full Outer Join
5. Cross Join
6. Semi Join
7. Anti Join

Basic Syntax of **merge()** function in R:

> ***Syntax:*** *merge(df1, df2, by.df1, by.df2, all.df1, all.df2, sort = TRUE)*
> ***Parameters: df1:*** *one dataframe* ***df2:*** *another dataframe* ***by.df1, by.df2:*** *The names of the columns that are common to both df1 and df2.* ***all, all.df1, all.df2:*** *Logical values that actually specify the type of merging happens.*

Python3

```
df1 = data.frame(StudentId = c(101:106),
                 Product = c("Hindi", "English",
                             "Maths", "Science",
                             "Political Science",
                             "Physics"))
df1
```

Output:

```
  StudentId          Product
1       101            Hindi
2       102          English
3       103            Maths
4       104          Science
5       105 Political Science
6       106          Physics
```

 # Data frame 2

Python3

```
df2 = data.frame(StudentId = c(102, 104, 106,
                               107, 108),
                 State = c("Mangalore", "Mysore",
                           "Pune", "Dehradun", "Delhi"))
df2
```

Output:

```
  StudentId   State
1       102 Mangalore
2       104   Mysore
3       106     Pune
4       107 Dehradun
5       108    Delhi
```

this, we actually specify the argument **all = FALSE**. If we try to understand this using set theory then we can say here we are actually performing the intersection operation. For example:

It is the most simplest and common type of joins available in R. Now let us try to understand this using R program:

Example:

Python3

```
# R program to illustrate
# Joining of dataframes

df = merge(x = df1, y = df2, by = "StudentId")
df
```

Output:

```
   StudentId Product    State
1        102 English Mangalore
2        104 Science   Mysore
3        106 Physics     Pune
```

Left Outer Join

Left Outer Join is basically to include all the rows of your dataframe x and only those from y that match, in this, we actually specify the argument **x = TRUE**. If we try to understand this using a basic set theory then we can say here we are actually displaying complete set x. Now let us try to understand this using R program:

Example:

Python3

```
df = merge(x = df1, y = df2, by = "StudentId",
                                all.x = TRUE)
df
```

Output:

```
  StudentId           Product    State
1       101             Hindi     NA
2       102           English   Mangalore
3       103             Maths     NA
4       104           Science   Mysore
5       105 Political Science     NA
6       106           Physics     Pune
```

Right Outer Join

Right, Outer Join is basically to include all the rows of your dataframe y and only those from x that match, in this, we actually specify the argument **y = TRUE**. If we try to understand this using a basic set theory then we can say here we are actually displaying a complete set y. Now let us try to understand this using R program:

Example:

Python3

```
# R program to illustrate
# Joining of dataframes

df = merge(x = df1, y = df2, by = "StudentId",
                                all.y = TRUE)
df
```

Output:

```
  StudentId Product    State
1       102 English Mangalore
2       104 Science   Mysore
3       106 Physics     Pune
```

Full Outer Join

Outer Join is basically used to keep all rows from both dataframes, in this, we actually specify the arguments **all = TRUE**. If we try to understand this using a basic set theory then we can say here we are actually performing the union option. Now let us try to understand this using R program:

Example:

Python3

```
# R program to illustrate
# Joining of dataframes

df = merge(x = df1, y = df2, by = "StudentId",
                                    all = TRUE)
df
```

Output:

```
   StudentId            Product    State
1        101              Hindi     NA
2        102            English  Mangalore
3        103              Maths     NA
4        104            Science   Mysore
5        105 Political Science     NA
6        106            Physics    Pune
7        107               NA     Dehradun
8        108               NA     Delhi
```

Cross Join

A Cross Join also known as cartesian join results in every row of one dataframe is being joined to every other row of another dataframe. In set theory, this type of joins is known as the cartesian product between two sets. Now let us try to understand this using R program:

Example:

```
# R program to illustrate
# Joining of dataframes

df = merge(x = df1, y = df2, by = NULL)
df
```

Output:

StudentId.x	Product	StudentId.y	State
1	101	Hindi	102	Mangalore
2	102	English	102	Mangalore
3	103	Maths	102	Mangalore
4	104	Science	102	Mangalore
5	105	Political Science	102	Mangalore
6	106	Physics	102	Mangalore
7	101	Hindi	104	Mysore
8	102	English	104	Mysore
9	103	Maths	104	Mysore
10	104	Science	104	Mysore
11	105	Political Science	104	Mysore
12	106	Physics	104	Mysore
13	101	Hindi	106	Pune
14	102	English	106	Pune
15	103	Maths	106	Pune
16	104	Science	106	Pune
17	105	Political Science	106	Pune
18	106	Physics	106	Pune
19	101	Hindi	107	Dehradun
20	102	English	107	Dehradun
21	103	Maths	107	Dehradun
22	104	Science	107	Dehradun
23	105	Political Science	107	Dehradun
24	106	Physics	107	Dehradun
25	101	Hindi	108	Delhi
26	102	English	108	Delhi
27	103	Maths	108	Delhi
28	104	Science	108	Delhi

Semi Join

This join is somewhat like inner join, with only the left dataframe columns and values are selected. Now let us try to understand this using R program:

Example:

Python3

```
# R program to illustrate
# Joining of dataframes

# Import required library
library(dplyr)

df = df1 %>% semi_join(df2, by = "StudentId")
df
```

Output:

```
  StudentId Product
1       102 English
2       104 Science
3       106 Physics
```

Anti Join

In terms of set theory, we can say anti-join as set difference operation, for example, A = (1, 2, 3, 4) B = (2, 3, 5) then the output of A-B will be set (1, 4). This join is somewhat like df1 – df2, as it basically selects all rows from df1 that are actually not present in df2. Now let us try to understand this using R program:

Example:

Python3

```
# R program to illustrate
# Joining of dataframes

# Import required library
library(dplyr)
```

Output:

```
   StudentId           Product
1        101             Hindi
2        103             Maths
3        105 Political Science
```

Next

R - DataFrame Manipulation

RECOMMENDED ARTICLES Page : 1 2 3

01 Joining Data in R with Dplyr Package
25, Nov 21

02 R - Merge Multiple DataFrames in List
07, Oct 21

03 How To Merge Two DataFrames in R ?
24, Mar 21

05 How to merge R DataFrames of different length ?
01, Apr 21

06 Combine vectors or DataFrames of unequal length in R
05, Apr 21

07 How to find the difference between two dataframes in R ?
15, Apr 21

28, Apr 20 27, Apr 21

Article Contributed By :

Akashkumar17
@Akashkumar17

Vote for difficulty

Current difficulty : Basic

Easy | Normal | Medium | Hard | Expert

Improved By : rkbhola5

Article Tags : Picked, R-DataFrame, R Language

Improve Article | Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

GeeksforGeeks
A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org