How can I use the coalesce() function in dplyr, and what are some examples of its usage?

Authored by

stats writer

June 27, 2024

The coalesce() function in dplyr is a useful tool for handling missing or NULL values in data frames. It allows users to select the first non-missing value from a set of columns, and can be used in various scenarios such as data cleaning and data analysis. To use the coalesce() function, users need to specify the columns they want to check for non-missing values. If all values in those columns are missing, the function will return NULL. Some examples of its usage include filling in missing values with the first non-missing value from a set of columns, creating a new column with the first non-missing value from a set of columns, and calculating summary statistics while ignoring missing values. Overall, the coalesce() function is a helpful tool for managing missing data and ensuring accurate analysis.

Use the coalesce() Function in dplyr (With Examples)

You can use the coalesce() function from the package in R to return the first non-missing value in each position of one or more vectors.

There are two common ways to use this function:

Method 1: Replace Missing Values in Vector

library(dplyr)

#replace missing values with 100
coalesce(x, 100)

Method 2: Return First Non-Missing Value Across Data Frame Columns

library(dplyr)

#return first non-missing value at each position across columns A and B
coalesce(dfA, dfB)

The following examples show how to each method in practice.

Example 1: Use coalesce() to Replace Missing Values in Vector

The following code shows how to use the coalesce() function to replace all missing values in a vector with a value of 100:

library(dplyr)

#create vector of values
x <- c(4, NA, 12, NA, 5, 14, 19)

#replace missing values with 100
coalesce(x, 100)

4 100 12 100 5 14 19

Notice that each NA value in the original vector has been replaced with a value of 100.

Example 2: Use coalesce() to Return First Non-Missing Value Across Data Frame Columns

Suppose we have the following data frame in R:

```
#create data frame
df <- data.frame(A=c(10, NA, 5, 6, NA, 7, NA),
B=c(14, 9, NA, 3, NA, 10, 4))

#view data frame
df
```

A B
1 10 14
2 NA 9
3 5 NA
4 6 3
5 NA NA
6 7 10
7 NA 4

The following code shows how to use the coalesce() function to return the first non-missing value across columns A and B in the data frame:

```
library(dplyr)
```

#create new column that coalesces values from columns A and B

df$C <- coalesce(df$A, df$B)

#view updated data frame

df

A B C

1 10 14 10

2 NA 9 9

3 5 NA 5

4 6 3 6

5 NA NA NA

6 7 10 7

7 NA 4 4

The resulting column C contains the first non-missing value across columns A and B.

We can simply add one more value to the coalesce() function to use as the value if there happen to be NA values in each column:

library(dplyr)

```
#create new column that coalesces values from columns A and B
df$C <- coalesce(df$A, df$B, 100)

#view updated data frame
df
```

```
A B C
1 10 14 10
2 NA 9 9
3 5 NA 5
4 6 3 6
5 NA NA 100
6 7 10 7
7 NA 4 4
```

Notice that the NA value in row 5 of column C has now been replaced by a value of 100.

The following tutorials explain how to perform other common functions using dplyr: