How can I replace all instances of zero (0) with NA in a specific column of a dataframe in R?

Authored by

stats writer

June 24, 2024

This is a simple and straightforward process that can be done in just a few steps using the "replace" function in R. First, identify the specific column in your dataframe where you want to replace the zeros. Then, use the replace function to replace all instances of zero with "NA" in that column. This will ensure that any data points with a value of zero will be replaced with "NA" instead. This can be useful for data cleaning and analysis, as it allows you to easily identify and handle missing or invalid data in that specific column. With this method, you can efficiently replace all instances of zero with "NA" in a specific column of your dataframe in R.

How to replace zero with NA on the data frame columns of single/multiple? You can use R base df notation to replace zero with NA value. NA is used in R to indicate missing values, which is different from 0 or empty strings(''). In this article, I will explain how to replace all occurrences of zeros with NAs on single/multiple columns of a data frame using the following approaches of R.

Create a data frame with numeric columns which has zero(0) values.

```
# Create dataframe with numeric columns
df = data.frame(pages=c(32,0,0),
chapters=c(20,86,0),
price=c(144,0,321))
print("Create data frame:")
print(df)
```

Yields below output.

```
[1] "Create data frame:"

> print(df)
  pages chapters price
1    32       20   144
2     0       86     0
3     0        0   321
```

Replace 0s with NAs in an R

To replace `0` with `NA` in a dataframe using the first R approach, which is the `df` notation, you can pass the condition (`df == 0`) into the `df` notation. This will check if the column value in the dataframe is 0, and if it is, it will be replaced with the NA value.

You cal also replace NA with 0 in R .

```
# Replace zeros with NA on data frame
df <- NA
print("After replacing zeros with NA:")
print(df)
```

Yields below output.

```
[1] "After replacing zeros with NA:"

> print(df)
  pages chapters price
1    32       20   144
2    NA       86    NA
3    NA       NA   321
```

Replace 0s with NAs in Selected Columns

Alternatively, you can use df notation to replace the zeros(0) with NA values on selected columns of the data frame. For this, you need to check if each value of the specified column is equal to 0. If it is 0, you can replace it with NA.

```
#Example 3 - Replace on selected columns
df == 0] <- NA
print(df)
```

Output
pages chapters price
1 32 20 144
2 NA 86 0
3 NA 0 321

Replace using Negation

You can also achieve this using the negation operator and `is.na()` function.

```
# Replace using negation
is.na(df) <- !df
print("After replacing zeros with NA:")
```

```
print(df)
```

Yields the same output as above

Using the R replace() function to Replace 0s with NAs

To replace all occurrences of zero values with NA values in a data frame, you can use the R replace() function. Simply pass the data frame along with the condition (representing the values to be replaced) and NA (representing the new values to replace with) into the function. This will give you a new data frame where all the zeros have been replaced with NAs.

```
#Example 4 - Using replace() function
df <- replace(df, df==0, NA)
print(df)
```

Output
pages chapters price
1 32 20 144
2 NA 86 NA
3 NA NA 321

Replace 0s with NAs using R dplyr

So far, we have implemented the replacing process using R base functions, Now we will see how to implement the replacing using the third-party package of R which is the dplyr package.

You can use the `na_if()` method to replace all zeros with NAs in an entire data frame. Before going to use the functions of the dplyr package we need to install it as `install.packages('dplyr')` and load it using `library("dplyr")`.

```
# Replace using dplyr::na_if()
library("dplyr")
df <- na_if(df, 0)
print(df)
```

Output
pages chapters price
1 32 20 144
2 NA 86 NA

3 NA NA 321

Replace 0s with NAs using dplyr::mutate_all()

To replace all presence of zeros in an entire data frame with NA you can use another function of the dplyr package is `mutate_all()` function. Let's apply it to a data frame and get the updated data frame where all zeros are substituted with NAs.

```
# Replace using dplyr::mutate_all()
library(dplyr)
df <- df %>% mutate_all(~na_if(., 0))
print(df)
```

Output
pages chapters price
1 32 20 144
2 NA 86 NA
3 NA NA 321

Replace on All Numeric columns

The `mutate_if()` function of the dplyr package is used to replace all occurrences of zeros only on numeric columns of the data frame with NA values. Since we have all numeric columns, it updates all columns of zero values with NA values.

```
# Replace only on all Numeric columns
library(dplyr)
df <- df %>% mutate_if(is.numeric, ~na_if(., 0))
print(df)
```

Output
pages chapters price
1 32 20 144
2 NA 86 NA
3 NA NA 321

Replace Zero with NA Only on Selected Columns

`mutate_at()` from `dplyr` is used to target specific columns (`pages`) in the dataframe `df` and replace 0 with NA.

```
# Replace only on selected columns
library(dplyr)
df <- df %>% mutate_at(c('pages'), ~na_if(., 0))
print(df)
```

Output
pages chapters price
1 32 20 144
2 NA 86 0
3 NA 0 321

Replace Zero with NA on the Selected Column Indexs

`mutate_at()` function from the dplyr package to replace all zeros with NAs on specified index position columns in the R dataframe.

```
# Replace only on selected column index
library(dplyr)
df <- df %>% mutate_at(c(2), ~na_if(., 0))
print(df)
```

Output
pages chapters price
1 32 20 144
2 0 86 0
3 0 NA 321

Replacing on tibble

If you have tibble, use the following approach to replace it. Use `tibble()` to create a tibble.

```
# Replacing on tibble
df <-tibble(
```

```
col1 = c("A", B, "NA"),
col2 = c(0, 2, NA),
col3 = c(1, NA, 5)
)
df <- df %>% mutate_if(is.numeric , replace_na, replace = 0)
print(df)
```

Output
A tibble: 3 × 3
col1 col2 col3
<chr> <dbl> <dbl>
1 A 0 1
2 B 2 0
3 NA 0 5

Conclusion

In this article, I have explained how to replace all occurrences of zeros with NAs on single/multiple columns of a data frame using multiple approaches of R such as df notation, replace(), and dplyr package functions like `na_if()`, `mutate_all()`, `mutate_if()`, and `mutate_at()`.

Related Articles

References