How can I replace NA values with 0 (zero) using the -R function?

Authored by

stats writer

June 24, 2024

The -R function is a tool that allows users to replace NA (not available) values with 0 (zero) in a dataset. This function is commonly used in data analysis and manipulation to fill in missing values with a placeholder, thus allowing for more accurate calculations and visualizations. By using the -R function, users can easily ensure that their data is complete and can be properly analyzed. This function can be applied to various types of data, such as numerical, categorical, or textual, making it a versatile and useful tool for data management.

How do I replace NA values on a numeric column with 0 (zero) in an R DataFrame (data.frame)? You can replace NA values with zero(0) on numeric columns of R data frame by using `is.na()`, `replace()`, `imputeTS::replace()`, `dplyr::coalesce()`, `dplyr::mutate_at()`, `dplyr::mutate_if()`, and `tidyr::replace_na()` functions.

It is best to replace numeric columns with zero or any value that makes sense, and for strings, replace them with empty space. Using these methods you can also replace NA values with empty string.

Generally, NA values are considered missing values, and doing any operation on these values results in inconsistent results, hence before processing data, it is good practice to handle these missing values. In this article, we will see how to replace NA values with Zero in an R data frame with examples like replaced by a single index, multiple indexes, single column name, multiple column names, and on all columns.

1. Quick Examples of Replace NA Values with 0

Below are quick examples of how to replace data frame column values from NA to 0 in R.

```
# Quick Examples of replace NA values with 0

# Example 1 - Replace na values with 0 using is.na()
my_dataframe <- 0

# Example 2 - Replace on selected column
my_dataframe)] <- 0
print(df)

# Example 3 - By using replace() & is.na()
my_dataframe <- replace(my_dataframe, is.na(my_dataframe), 0)

# Example 4 - Another way
my_dataframe <- my_dataframe %>% replace(is.na(.), 0)
```

```
# Example 5 - Load the imputeTS package
library("imputeTS")
# Replace NA avalues with 0
my_dataframe <- na_replace(my_dataframe, 0)

#Example 6 - Replace NA with zero on all numeric column
library("dplyr")
my_dataframe <- mutate_all(my_dataframe, ~coalesce(.,0))

# All below examples required these libraries
library("tidyr")
library("dplyr")

# Example 7 - Replace NA with zero on all numeric column
my_dataframe <- mutate_all(my_dataframe, ~replace_na(.,0))

# Example 8 - Replace NA using setnafill() from data.table
library("data.table")
my_dataframe <- setnafill(my_dataframe, fill=0)

# Example 9 - Replace na with zero on specific numeric column
# Load dplyr library
my_dataframe <- my_dataframe %>%
mutate(id = coalesce(id, 0))

# Example 10 - Replace on multiple columns
my_dataframe <- my_dataframe %>%
mutate(id = coalesce(id, 0),
pages = coalesce(pages, 0))

# Example 11 - Load tidyr library
my_dataframe <- my_dataframe %>%
mutate_at(1, ~replace_na(.,0))

# Example 12 - Replace NA on multiple columns by Index
my_dataframe <- my_dataframe %>%
mutate_at(c(1,3), ~replace_na(.,0))

# Example 13 - Replace NA on multiple columns by name
my_dataframe <- my_dataframe %>%
mutate_at(c('id','pages'), ~replace_na(.,0))
```

# Example 14 - Replace only numeric columns
my_dataframe <- my_dataframe %>%
mutate_if(is.numeric, ~replace_na(., 0))

As you noticed above, I have used the following methods to replace NA values with 0 in R.

Let's create a data frame with some NA values, run these examples, and validate the result.

```
# Create dataframe with 5 rows and 3 columns
my_dataframe=data.frame(id=c(2,1,3,4,NA),
name=c('sravan',NA,'chrisa','shivgami',NA),
gender=c(NA,'m',NA,'f',NA))
```

# Display dataframe
print(my_dataframe)

Output:

```
  id     name gender
1  2   sravan   <NA>
2  1     <NA>      m
3  3   chrisa   <NA>
4  4 shivgami      f
5 NA     <NA>   <NA>
```

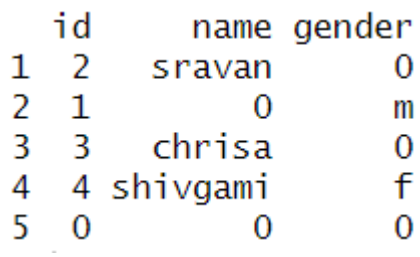
2. Replace NA values with 0 using is.na()

`is.na()` is used to check whether the given data frame column value is equal to NA or not in R. If it is NA, it will return the logical matrix of the same length as the given dataframe where `TRUE` for every NA value and `FALSE` for every non-NA values. So by specifying it inside- (index), it will return NA and assign it to 0. In this way, we can replace NA values with Zero(0) in an R DataFrame.

```
# Replace na values with 0 using is.na()
my_dataframe = 0
```

# Display the dataframe
print(my_dataframe)

Output:



	id	name	gender
1	2	sravan	0
2	1	0	m
3	3	chrisa	0
4	4	shivgami	f
5	0	0	0

In the above output, we can see that NA values are replaced with 0's.

Alternatively, you can use the is.na() function to replace the specific column of NA values with 0s in the R data frame. for example,

```
# Replace NA values of specific column with 0s
my_dataframe$name = 0
print(my_dataframe)
```

Output:
id name gender
1 2 sravan <NA>
2 1 0 m
3 3 chrisa <NA>
4 4 shivgami f
5 NA 0 <NA>

3. Replace NA values with 0 in a DataFrame using replace()

Let's see another way to change NA values with zero using the `replace()`. It will take three parameters.

```
# Replace NA avalues with 0
my_dataframe <- replace(my_dataframe,is.na(my_dataframe),0)
```

The first parameter is the input data frame.The second parameter takes `is.na()` method to check if it is NAThe last parameter takes the value 0, which will replace the value present in the second parameter

Output:

```
# Output
id name gender
1 2 sravan 0
2 1 0 m
3 3 chrisa 0
4 4 shivgami f
5 0 0 0
```

In the above output, we can see that NA values are replaced with 0's.

4. Replace NA values with 0 using replace() from "imputeTS"

`replace()` is used to replace NA with 0 in an R data frame. It is available in `imputeTS` package. So we have to install and load this package before using the rename() method.

`imputeTS` is a third-party library hence, to use imputeTS library, you need to first install it by using install.packages('imputeTS'). Once installation is completed, load the imputeTS library to use this `replace()` method. To load a library in R, use `library("imputeTS")`.

```
#Replace NA avalues with 0
my_dataframe <- na_replace(my_dataframe, 0)
```

Output:

```
# Output
id name gender
1 2 sravan 0
2 1 0 m
3 3 chrisa 0
4 4 shivgami f
5 0 0 0
```

In the above output, we can see that NA values are replaced with 0's.

5. Replace NA with Zero on All Numeric Values

There are several other ways to rename NA with zero in the R data frame by using methods from the dplyr package.

All previous examples use the Base R built-in functions that can be used on a smaller dataset but, for bigger data sets, you have to use methods from `dplyr` package as they perform 30% faster. `dplyr` package uses C++ code to evaluate. Let's create another data frame with all numeric columns and run these examples.

```
# Create dataframe with numeric columns
my_dataframe=data.frame(pages=c(32,45,NA,22,NA),
chapters=c(NA,86,11,15,NA),
price=c(144,553,321,567,NA))

# Replace NA using coalesce() from dplyr
library("dplyr")
my_dataframe <- mutate_all(my_dataframe, ~coalesce(.,0))

# Replace NA using replace_na() from tidyr
library("dplyr")
library("tidyr")
my_dataframe <- mutate_all(my_dataframe, ~replace_na(.,0))

# Replace NA using setnafill() from data.table
library("data.table")
my_dataframe <- setnafill(my_dataframe, fill=0)
```

All the above examples yield the same below output.

```
# Output
id pages chapters price
1 11 32 0 144
2 22 45 86 553
3 33 0 11 321
4 44 22 15 567
5 0 0 0 0
```

Here, the `coalesce()` function is from `dplyr` package. This returns the first non-missing value of

its arguments.

6. Update NA with Zero By Specific Column Name

Here we can use the `mutate()` function along with `coalesce()` from `dplyr` package. This updates NA values with zero on the `id` column. By using this on character columns you will get an error.

```
# Load dplyr library
library("dplyr")
#Replace NA with zero on specific numeric column
my_dataframe <- my_dataframe %>%
mutate(id = coalesce(id, 0))
```

7. Update NA with Zero on Multiple Columns by Name

Let's use the same above approach but replace NA with zero on multiple columns by column name.

```
# Replace on multiple columns
library("dplyr")
my_dataframe <- my_dataframe %>%
mutate(id = coalesce(id, 0),
pages = coalesce(pages, 0))
```

8. Replace NA with 0 on Column by Index

Use `mutate_at()` to specify the index number where you wanted to replace NA values with zero in R data frame.

```
# Load tidyr library
library("tidyr")
library("dplyr")
my_dataframe <- my_dataframe %>%
mutate_at(1, ~replace_na(.,0))
print(my_dataframe)
```

Yields below output.

```
# Output
id pages chapters price
1 11 32 NA 144
2 22 45 86 553
3 33 NA 11 321
4 44 22 15 567
5 0 NA NA NA
```

9. Replace NA on Multiple Columns by Index

`mutate_at()` also takes a vector with index numbers which is used to replace NA with 0 on multiple columns and `replace_na()` replaces all NA with 0.

```
# Replace NA on multiple columns by Index
library("tidyr")
library("dplyr")
my_dataframe <- my_dataframe %>%
mutate_at(c(1,3), ~replace_na(.,0))
print(my_dataframe)
```

Yields below output.

```
# Output
id pages chapters price
1 11 32 0 144
2 22 45 86 553
3 33 NA 11 321
4 44 22 15 567
5 0 NA 0 NA
```

10. Replace Only on Numeric Columns

When you have the data.frame with a mix of numeric and character columns, to update only numeric columns from NA with 0 use `mutate_if()` with `is.numeric` as a parameter.

```
# Replace only numeric columns
library("tidyr")
library("dplyr")
```

```
my_dataframe <- my_dataframe %>%
mutate_if(is.numeric, ~replace_na(., 0))
```

11. Data with Factor Values

If you have data with numeric and characters most of the above examples work without issue. But, if you have factor values, first you need to convert them to a character before replacing NA with zero.

```
#Change factors to character type
my_dataframe <- lapply(my_dataframe, as.character)

# Replace NA with 0
my_dataframe <- 0

# Change character columns back to factors
my_dataframe <- lapply(my_dataframe, as.factor)
```

Frequently Asked Questions of Replace NA values with 0 in R

How can I replace NA values with 0 in a specific column of a data frame?

To replace NA values with 0 in a specific column of a data frame in R, you can use the below code. For example, `df$specific_column_name <- 0`.

How do I replace all NA values in a data frame with 0?

You can use the is.na() function to replace all NA values in a data frame. For example, `df = 0`.

12. Conclusion

In this article, I have explained several ways to replace NA values with zero (0) on numeric columns of the R data frame. We can use the `replace()` method in two ways. One is from the `imputeTS` package and another way is we can use it directly.

Related Articles

References

replace() in RimputeTS() package in RNA