

How can I remove duplicate rows in R?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I remove duplicate rows in R?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=149846>

To remove duplicate rows in R, one can use the function "duplicated()" which identifies and returns a logical vector indicating which rows are duplicates. This vector can then be used to subset the data frame and remove the duplicate rows. Alternatively, one can use the function "distinct()" from the dplyr package, which returns a data frame with only unique rows. Both methods provide efficient and effective ways to remove duplicate rows in R.

To remove duplicates or duplicate rows in R DataFrame (data.frame)? There are multiple ways to get the duplicate rows in R by removing all duplicates from a single column, selected columns, or all columns. In this article, I will explain all these examples by using functions from R base, dplyr, and data.table.

1. Quick Examples of Remove Duplicate Rows

Below are quick examples of removing duplicates or duplicate rows from R data frame (data.frame)

```
# Quick Examples
```

```
# Remove duplicate rows
```

```
df2 <- df
```

```
# Remove duplicates by single column
```

```
df2 <- df
```

```
# Remove duplicates on selected columns
```

```
df2 <- unique( df )
```

```
# Using dplyr
```

```
# Remove duplicate rows (all columns)
```

```
library(dplyr)
```

```
df2 <- df %>% distinct()
```

```
# Remove duplicates on specific column
```

```
df2 <- df %>% distinct(id, .keep_all = TRUE)
```

```
# Remove duplicates on selected columns
```

```
df2 <- df %>% distinct(id,pages, .keep_all = TRUE)
```

```
# using data.table
```

```
library(data.table)
```

```
dt <- data.table(df)
```

```
#Remove duplicates on specific column
```

```
dt2 <- unique(dt, by = "id")
```

Let's create an R DataFrame,

```
# Create dataframe
df=data.frame(id=c(11,11,33,44,44),
pages=c(32,32,33,22,22),
name=c("spark","spark","R","java","jsp"),
chapters=c(76,76,11,15,15),
price=c(144,144,321,567,567))
df
```

Yields below output. Note that the first 2 rows have duplicates (all column values) and the last two rows have duplicates on columns of `id`, `pages`, `chapters` and `price`.

```
# Output
id pages name chapters price
1 11 32 spark 76 144
2 11 32 spark 76 144
3 33 33 R 11 321
4 44 22 java 15 567
5 44 22 jsp 15 567
```

2. Remove Duplicates using R Base Functions

R base provides `duplicated()` and `unique()` functions to remove duplicates in an R DataFrame (`data.frame`). By using these two functions we can delete duplicate rows by considering all columns, single column, or selected columns.

2.1 Remove Duplicate Rows

In R, `duplicated()` function that takes vector or `data.frame` as input and selects rows that are duplicates, by negating the result you will remove all duplicate rows in the R `data.frame`. For example, from my data frame above having the first 2 rows are duplicated, running the below example eliminates duplicate rows and returns the data frame with unique rows.

```
# Remove duplicate rows
df2 <- df
```

```
df2
```

```
# Output
```

```
# id pages name chapters price
```

```
# 1 11 32 spark 76 144
```

```
# 3 33 33 R 11 321
```

```
# 4 44 22 java 15 567
```

```
# 5 44 22 jsp 15 567
```

In case you want to remove duplicates based on a single column, use the column name as an argument to the function.

```
# Remove duplicates by single column
```

```
df2 <- df
```

```
df2
```

```
# Output
```

```
# id pages name chapters price
```

```
# 1 11 32 spark 76 144
```

```
# 3 33 33 R 11 321
```

```
# 4 44 22 java 15 567
```

2.2 Remove Duplicates on Selected Columns

Use the `unique()` function to remove duplicates from the selected multiple columns of the R data frame. The following example removes duplicates by selecting columns `id`, `pages`, `chapters` and `price`.

```
# Remove duplicates on selected columns
```

```
df2 <- unique( df )
```

```
df2
```

```
# Output
```

```
# id pages chapters price
```

```
# 1 11 32 76 144
```

```
# 3 33 33 11 321
```

```
# 4 44 22 15 567
```

3. Remove Duplicate Rows using the dplyr

`dplyr` package provides `distinct()` function to remove duplicates, to use this, you need to load the library using `library("dplyr")` to use its methods. In case you don't have this package, install it using `install.packages("dplyr")`.

For bigger data sets it is best to use the methods from the `dplyr` package as they perform 30% faster. the `dplyr` package uses C++ code to evaluate.

3.1 Use `distinct()` to Remove Duplicates

`distinct()` method selects unique rows from a data frame by removing all duplicates in R. This is similar to the R base `unique` function but, this performs faster when you have large datasets, so use this when you want better performance.

```
# Using dplyr
# Remove duplicate rows (all columns)
library(dplyr)
df2 <- df %>% distinct()
df2

# Output
# id pages name chapters price
# 1 11 32 spark 76 144
# 2 33 33 R 11 321
# 3 44 22 java 15 567
# 4 44 22 jsp 15 567
```

Here, we use the infix operator `%>%` from `magrittr`, it passes the left-hand side of the operator to the first argument of the right-hand side of the operator. For example, `x %>% f(y)` converted into `f(x, y)` so the result from the left-hand side is then "piped" into the right-hand side.

3.2 Remove Duplicates on Specific column

Similarly, you can also use this to get duplicate rows on a single column. Here, I am using an optional argument `.keep_all=TRUE` which keeps all variables in `.data`. If a combination of `...` is not distinct, this keeps the first row of values.

```
#Remove duplicates on specific column
```

```
df2 <- df %>% distinct(id, .keep_all = TRUE)
df2
```

```
# Output
```

```
# id pages name chapters price
```

```
# 1 11 32 spark 76 144
```

```
# 2 33 33 R 11 321
```

```
# 3 44 22 java 15 567
```

3.3 Get Unique Rows on Selected Columns

If you want to get unique rows on selected columns of the R data.frame, just pass the columns as arguments to this [distinct\(\)](#) function.

```
#Remove duplicates on selected columns
df2 <- df %>% distinct(id,pages, .keep_all = TRUE)
df2
```

```
# Output
```

```
# id pages name chapters price
```

```
# 1 11 32 spark 76 144
```

```
# 2 33 33 R 11 321
```

```
# 3 44 22 java 15 567
```

4. Remove Duplicate Rows using data.table

To eliminate duplicates, use the `unique()` function from the `data.table` package. The [data.table](#) package is designed for handling tabular data in the R programming language, offering an efficient `data.table` object. This object provides significantly better performance compared to the default `data.frame`.

```
# using data.table
library(data.table)
dt <- data.table(df)
#Remove duplicates on specific column
dt2 <- unique(dt, by = "id")
dt2
```

```
# Output
```

```
# id pages name chapters price  
#1: 11 32 spark 76 144  
#2: 33 33 R 11 321  
#3: 44 22 java 15 567
```

Frequently Asked Questions on Remove Duplicate Rows

How do I identify and count duplicate rows in a data frame in R?

You can use the `duplicated()` function to identify duplicates and `sum(duplicated(df))` to count them in a data frame `df`.

What function is commonly used to remove duplicate rows in R?

The `unique()` function is commonly used to remove duplicate rows from a data frame in R.

How can I remove duplicates based on specific columns in R?

You can use the `duplicated()` function with the `subset` parameter to check for duplicates based on specific columns, and then use this information to filter the data frame accordingly.

How do I keep the first occurrence of each unique row and remove subsequent duplicates in R?

You can use the `duplicated()` function with the `fromLast` parameter set to `TRUE` to mark duplicates from the end. Then, you can use boolean indexing to keep the first occurrence.

Conclusion

In this article, you have learned how to remove duplicates or duplicate rows in R by using the R base function `duplicated()`, `unique()` and using the `dplyr` package function `distinct()` and finally using the `unique()` function from `data.table`. If a performance matters use either function from the `dplyr` or `data.table`.

Related Articles

References