

How to select rows in R with examples?

Authored by
stats writer

June 23, 2024

RECOMMENDED CITATION

stats writer (2024). *How to select rows in R with examples?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=149395>

In R, rows can be selected from a dataset using various methods. The most common way is by using the square bracket notation, where the index numbers or logical expressions are used to specify the rows to be selected. For example, "df" will select the first five rows of the dataframe "df". Similarly, "df" will select all the rows where the values in the column "column" are greater than 10. Additionally, the "subset" function and the "filter" function from the dplyr package can also be used to select rows based on certain conditions. Overall, selecting rows in R can be achieved through different approaches, providing flexibility and convenience in data manipulation.

By using bracket notation on the R data frame (data.frame) we can select rows by column value, by index, by name, by condition, etc. You can also use the R base function subset() to get the same results. Besides these, R also provides another function `dplyr::filter()` to get the rows from the DataFrame. If you have data.table then use the function from it to achieve better performance.

Some of the methods explained in this article are also used to select columns from an R data frame.

1. Quick Examples of Selecting Rows

Below are brief examples of selecting rows from data frame (data.frame) in R

```
# Quick Examples of selecting rows
```

```
# Example 1: Select Rows by index
```

```
df
```

```
# Example 2: Select Rows by list of index values
```

```
df
```

```
# Example 3: Select Rows by index range
```

```
df
```

```
# Example 4: Select first N rows
```

```
head(df,3)
```

```
# Example 5: Select last N rows
```

```
tail(df,3)
```

```
# Example 6: Select Rows by name
```

```
df
```

```
# Example 7: Select Rows by list of names
```

```
df
```

```
# Example 8: Select Rows by column value
df

# Example 9: Select Rows by checking values on multiple columns
df

# Example 10: Select Rows by list of column values
df

# Example 11: Using is.element()
df

# Example 12: Using subset
subset(df, state %in% c("CA", "AZ", "PH"))

# Example 13: Using dplyr::filter
dplyr::filter(df, state %in% c("CA", "AZ", "PH"))

# Example 14: Using data.table
library(data.table)
setDT(df, key = 'state')
```

Let's create an R data frame,

```
# Create DataFrame
df <- data.frame(
  id = c(10,11,12,13,14,15,16,17),
  name = c('sai','ram','deepika','sahithi','kumar','scott','Don','Lin'),
  gender = c('M','M','F','F','M','M','M','F'),
  dob = as.Date(c('1990-10-02','1981-3-24','1987-6-14','1985-8-16',
'1990-10-02','1981-3-24','1987-6-14','1985-8-16')),
  state = c('CA','NY',NA,NA,'DC','DW','AZ','PH')
)
df
```

Yields below output.

	id	name	gender	dob	state
1	10	sai	M	1990-10-02	CA
2	11	ram	M	1981-03-24	NY
3	12	deepika	F	1987-06-14	<NA>
4	13	sahithi	F	1985-08-16	<NA>
5	14	kumar	M	1990-10-02	DC
6	15	scott	M	1981-03-24	DW
7	16	Don	M	1987-06-14	AZ
8	17	Lin	F	1985-08-16	PH

2. Using R base to Select Rows

You can use a bracket notation to select rows from DataFrame in R. In this selection, I will cover how to select rows by index, select rows by Name, and check column values. All these return a DataFrame after selecting the specific rows hence, you can use these to [Create an R DataFrame from the existing DataFrame](#)

2.1 By Index

Every row or observation in a DataFrame is assigned an index, you can use this index to get rows. Following are some commonly used methods to [select rows by index in R](#).

```
# Select Rows by Index  
df
```

```
# Select Rows by List of Index Values  
df
```

Yields below output.

```
> # Select Rows by Index
> df[3,]
  id   name gender      dob state
3 12 deepika      F 1987-06-14 <NA>

> # Select Rows by List of Index Values
> df[c(3,4,6),]
  id   name gender      dob state
3 12 deepika      F 1987-06-14 <NA>
4 13 sahithi      F 1985-08-16 <NA>
6 15  scott      M 1981-03-24   DW
```

Below are some more ways to select rows using an index of the R data frame. The first way is to get the specified portion of rows using the index range, the second way is to get the first N rows and the final way is to get the last N rows.

```
# Select Rows by Index Range
df
```

```
# Output:
```

```
# id name gender dob state
# 3 12 deepika F 1987-06-14 <NA>
# 4 13 sahithi F 1985-08-16 <NA>
# 5 14 kumar M 1990-10-02 DC
# 6 15 scott M 1981-03-24 DW
```

```
# Select first N rows
```

```
head(df,3)
```

```
# Output:
```

```
# id name gender dob state
# 1 10 sai M 1990-10-02 CA
# 2 11 ram M 1981-03-24 NY
# 3 12 deepika F 1987-06-14 <NA>
```

```
# Select last N rows
```

```
tail(df,3)
```

```
# Output:
```

```
# id name gender dob state
# 1 10 sai M 1990-10-02 CA
```

```
# 2 11 ram M 1981-03-24 NY
# 3 12 deepika F 1987-06-14 <NA>
```

2.2 By Row Name

If you have row names on the data frame and want to select rows by row name in R, use the below approach.

By default, row names in an R data frame are assigned as incremental sequence numbers during its creation. However, R allows for the assignment of custom row names either during the creation of the data frame or by using the `rownames()` function on an existing DataFrame. Similarly, to set the column names, the `colnames()` function can be used.

Create the R data frame by customizing the row names using `rownames()` function.

```
# Create R data frame
df <- data.frame(
  id = c(10,11,12,13,14,15,16,17),
  name = c('sai','ram','deepika','sahithi','kumar','scott','Don','Lin'),
  gender = c('M','M','F','F','M','M','M','F'),
  dob = as.Date(c('1990-10-02','1981-3-24','1987-6-14','1985-8-16',
    '1990-10-02','1981-3-24','1987-6-14','1985-8-16')),
  state = c('CA','NY',NA,NA,'DC','DW','AZ','PH')
)
# Set custom row names
custom_row_names <- c('r1','r2','r3','r4','r5','r6','r7','r8')
rownames(df) <- custom_row_names
print(df)
```

Yields below output.

```
# Output:
id name gender dob state
r1 10 sai M 1990-10-02 CA
r2 11 ram M 1981-03-24 NY
r3 12 deepika F 1987-06-14 <NA>
r4 13 sahithi F 1985-08-16 <NA>
r5 14 kumar M 1990-10-02 DC
r6 15 scott M 1981-03-24 DW
r7 16 Don M 1987-06-14 AZ
r8 17 Lin F 1985-08-16 PH
```

To get the single/multiple rows by name you can use the vector with the specified values, you want to return.

```
# Select Rows by Name
df

# Output:
# id name gender dob state
# r3 12 deepika F 1987-06-14 <NA>
```

```
# Select Rows by list of names
df
```

```
# Output:
# id name gender dob state
# r3 12 deepika F 1987-06-14 <NA>
# r6 15 scott M 1981-03-24 DW
```

2.3 By Checking Column Values

Let's see some examples of how to select rows by conditions in R, for example, conditions include equal, not equal. And also some examples to get rows based on multiple conditions. To get rows based on column value use `%in%` operator.

```
# Select Rows by equal condition
df
```

```
# Output:
# id name gender dob state
# r1 10 sai M 1990-10-02 CA
# r2 11 ram M 1981-03-24 NY
# r5 14 kumar M 1990-10-02 DC
# r6 15 scott M 1981-03-24 DW
# r7 16 Don M 1987-06-14 AZ
```

```
# Select Rows by not equal condition
df
```

```
# Output:
# id name gender dob state
```

```
# r3 12 deepika F 1987-06-14 <NA>
# r4 13 sahithi F 1985-08-16 <NA>
# r8 17 Lin F 1985-08-16 PH
```

```
# Select Rows by Multiple Conditions
df
```

```
# Output:
# id name gender dob state
# r7 16 Don M 1987-06-14 AZ
```

```
# Select rows based on list
df
```

```
# Output:
# id name gender dob state
# r4 13 sahithi F 1985-08-16 <NA>
# r5 14 kumar M 1990-10-02 DC
# r6 15 scott M 1981-03-24 DW
```

3. Select Rows with head() and tail()

Similarly, You can use the head() function to get the first N rows and use the tail() to get the last N rows. These two methods take a dataframe object as the first argument and an integer value as the second argument that specifies how many rows to return. For example,

```
# Select first N rows
head(df,3)
```

```
# Output:
# id name gender dob state
# r1 10 sai M 1990-10-02 CA
# r2 11 ram M 1981-03-24 NY
# r3 12 deepika F 1987-06-14 <NA>
```

```
# Select last N rows
tail(df,3)
```

```
# Output:
# id name gender dob state
```

```
# r6 15 scott M 1981-03-24 DW
# r7 16 Don M 1987-06-14 AZ
# r8 17 Lin F 1985-08-16 PH
```

4. Using subset() to Select Rows from DataFrame

The `subset()` function is an R base function commonly used to select specific rows from a data frame. It accepts a data frame as an argument, along with a condition that defines which rows you want to select.

```
# Using subset to select rows
subset(df, state %in% c("CA", "AZ", "PH"))
```

Output:

```
# id name gender dob state
# r1 10 sai M 1990-10-02 CA
# r7 16 Don M 1987-06-14 AZ
# r8 17 Lin F 1985-08-16 PH
```

5. Using filter() from dplyr Package

The <https://arabpsychology.com/r-programming/r-dplyr-tutorial-learn-with-examples/> `dplyr` package provides a structured approach to data manipulation. It includes a collection of popular functions to help data analysts with common data transformation tasks.

To work with `dplyr`, start by installing it with `install.packages('dplyr')` and then load it using `library(dplyr)`. In this example, I'll use the `dplyr::filter()` function from this package to select rows based on specific column values.

```
# Load dplyr package
library(dplyr)

# Using dplyr::filter
dplyr::filter(df, state %in% c("CA", "AZ", "PH"))
```

Output:

```
# id name gender dob state
# r1 10 sai M 1990-10-02 CA
```

```
# r7 16 Don M 1987-06-14 AZ  
# r8 17 Lin F 1985-08-16 PH
```

6. Using setDT() from data.table Package

The `data.table` package in R is designed for handling tabular data. It offers highly efficient `data.table` object, which is an optimized and high-performance alternative to the default `data.frame`.

To use the `data.table`, you need to install the package with `install.packages('data.table')` and load it using `library(data.table)`. In this example, I will use `data.table` methods to select specific rows based on the values in certain columns.

```
# Using data.table  
library(data.table)  
setDT(df, key = 'state')
```

Yields the same as the above output.

7. Conclusion

In this article, you have learned by using bracket notation on an R data frame you can select rows by name, index, column value, condition, etc. Similarly, the R base `subset()` function gives the same results. In addition to these, R also offers another package `dplyr` to return the selected rows from the data frame. If you have a `data.table` then use the function from it to achieve better performance.

Related Articles

References