

Package ‘sasif’

April 19, 2026

Type Package

Title 'SAS' IF Style Data Step Logic for Data Tables

Version 0.1.3

Description Provides 'SAS'-style IF/ELSE chains, independent IF rules, and DELETE logic for 'data.table', enabling clinical programmers to express Study Data Tabulation Model (SDTM) and Analysis Data Model (ADaM)-style derivations in familiar SAS-like syntax. Methods are informed by clinical data standards described in CDISC SDTM and ADaM implementation guides. See <https://www.cdisc.org/standards/foundational/sdtm> and <https://www.cdisc.org/standards/foundational/adam>.

License MIT + file LICENSE

Encoding UTF-8

URL <https://chandrt23-lang.github.io/sasif/>,
<https://github.com/chandrt23-lang/sasif>

BugReports <https://github.com/chandrt23-lang/sasif/issues>

Imports data.table

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Thiyagarajan Chandrasekaran [aut, cre]

Maintainer Thiyagarajan Chandrasekaran <chandrt23@gmail.com>

Repository CRAN

Date/Publication 2026-04-19 09:00:02 UTC

Contents

data_step	2
delete_if	3
else_do	3
else_if_do	4
if_do	4
if_independent	5
Index	6

data_step	<i>SAS IF-style data step logic for data.table</i>
-----------	--

Description

Provides SAS-style IF/ELSE chains, independent IF rules, and DELETE logic for fast, vectorized transformations on data.table objects. This enables clinical programmers to express SDTM and ADaM-style derivations in familiar SAS-like syntax while leveraging data.table performance.

Usage

```
data_step(dt, ..., copy = TRUE)
```

Arguments

dt	A data.table.
...	One or more rule objects created by if_do(), else_if_do(), else_do(), if_independent(), or delete_if().
copy	Logical. If TRUE (default), a copy of dt is modified and returned.

Value

A data.table with applied transformations.

Examples

```
library(data.table)

dt <- data.table(
  AGE = c(40, 60, 80),
  SEX = c("M", "F", "M")
)

out <- data_step(
  dt,
  if_do(AGE <= 45, GROUP = 1),
  else_if_do(AGE <= 70, GROUP = 2),
  else_do(GROUP = 3),
)
```

```
    if_independent(SEX == "M", MALE = 1)
  )
out
```

delete_if	<i>Create a SAS-style DELETE rule</i>
-----------	---------------------------------------

Description

Creates a DELETE rule to remove rows from the data.table when condition is TRUE.

Usage

```
delete_if(condition)
```

Arguments

condition Logical condition evaluated on the data.table.

Value

A rule object for data_step().

else_do	<i>Create a SAS-style ELSE rule</i>
---------	-------------------------------------

Description

Creates an ELSE rule for use inside data_step().

Usage

```
else_do(...)
```

Arguments

... Named assignments to apply when no previous IF/ELSE IF matched.

Value

A rule object for data_step().

else_if_do	<i>Create a SAS-style ELSE IF rule</i>
------------	--

Description

Creates an ELSE IF rule for use inside data_step().

Usage

```
else_if_do(condition, ...)
```

Arguments

condition	Logical condition evaluated on the data.table.
...	Named assignments to apply when condition is TRUE.

Value

A rule object for data_step().

if_do	<i>Create a SAS-style IF rule</i>
-------	-----------------------------------

Description

Creates a mutually exclusive IF rule for use inside data_step().

Usage

```
if_do(condition, ...)
```

Arguments

condition	Logical condition evaluated on the data.table.
...	Named assignments to apply when condition is TRUE.

Value

A rule object for data_step().

if_independent	<i>Create an independent SAS-style IF rule</i>
----------------	--

Description

Creates an independent IF rule that is evaluated regardless of IF/ELSE chains.

Usage

```
if_independent(condition, ...)
```

Arguments

condition	Logical condition evaluated on the data.table.
...	Named assignments to apply when condition is TRUE.

Value

A rule object for data_step().

Index

data_step, 2
delete_if, 3

else_do, 3
else_if_do, 4

if_do, 4
if_independent, 5