Package ‘sasif’

February 3, 2026

Type Package

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

Version 0.1.2

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

Imports data.table

RoxygenNote 7.3.3

NeedsCompilation no

Author Thiyagarajan Chandrasekaran [aut, cre]

Maintainer Thiyagarajan Chandrasekaran <chandrt23@gmail.com>

Repository CRAN

Date/Publication 2026-02-03 12:40:02 UTC

Contents

data_step *SAS IF-style data step logic for data.table*

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` *Create a SAS-style DELETE rule*

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` *Create a SAS-style ELSE rule*

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 *Create a SAS-style ELSE IF rule*

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 *Create a SAS-style IF rule*

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` *Create an independent SAS-style IF rule*

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