

R documentation

of ‘if_else.Rd’

May 13, 2025

if_else

Vectorised if-else

Description

`if_else()` is a vectorized [if-else](#). Compared to the base R equivalent, `ifelse()`, this function allows you to handle missing values in the condition with `missing` and always takes `true`, `false`, and `missing` into account when determining what the output type should be.

Usage

```
if_else(condition, true, false, missing = NULL, ..., ptype = NULL, size = NULL)
```

Arguments

<code>condition</code>	A logical vector
<code>true, false</code>	Vectors to use for TRUE and FALSE values of condition. Both <code>true</code> and <code>false</code> will be recycled to the size of <code>condition</code> . <code>true</code> , <code>false</code> , and <code>missing</code> (if used) will be cast to their common type.
<code>missing</code>	If not <code>NULL</code> , will be used as the value for NA values of <code>condition</code> . Follows the same size and type rules as <code>true</code> and <code>false</code> .
<code>...</code>	These dots are for future extensions and must be empty.
<code>ptype</code>	An optional prototype declaring the desired output type. If supplied, this overrides the common type of <code>true</code> , <code>false</code> , and <code>missing</code> .
<code>size</code>	An optional size declaring the desired output size. If supplied, this overrides the size of <code>condition</code> .

Value

A vector with the same size as condition and the same type as the common type of true, false, and missing.

Where condition is TRUE, the matching values from true, where it is FALSE, the matching values from false, and where it is NA, the matching values from missing, if provided, otherwise a missing value will be used.

Examples

```
x <- c(-5:5, NA)
if_else(x < 0, NA, x)

# Explicitly handle `NA` values in the `condition` with `missing`
if_else(x < 0, "negative", "positive", missing = "missing")

# Unlike `ifelse()`, `if_else()` preserves types
x <- factor(sample(letters[1:5], 10, replace = TRUE))
ifelse(x %in% c("a", "b", "c"), x, NA)
if_else(x %in% c("a", "b", "c"), x, NA)

# `if_else()` is often useful for creating new columns inside of `mutate()`
starwars %>%
  mutate(category = if_else(height < 100, "short", "tall"), .keep = "used")
```

Index

`if-else`, [/](#)
`if_else`, [1](#)
`ifelse()`, [/](#)
`recycled`, [/](#)