Package ‘fauxnaif’

October 13, 2022

Title Convert Values to NA
Version 0.7.1
Description Provides a replacement for dplyr::na_if(). Allows you to specify multiple values to be replaced with NA using a single function.
License MIT + file LICENSE
BugReports https://github.com/rossellhayes/fauxnaif/issues
Depends R (>= 3.5)
Imports cli, rlang (>= 1.0.0)
Suggests covr, dplyr, haven, knitr, magrittr, rmarkdown, testthat (>= 3.0.0), tibble, tidyr, vctrs, withr
VignetteBuilder knitr
Config/testthat/edition 3
Encoding UTF-8
Language en-US
LazyData true
RoxygenNote 7.2.1
NeedsCompilation no
Author Alexander Rossell Hayes [aut, cre, cph]
  (<https://orcid.org/0000-0001-9412-0457>)
Maintainer Alexander Rossell Hayes <alexander@rossellhayes.com>
Repository CRAN
Date/Publication 2022-08-12 17:30:02 UTC

R topics documented:

  faux_census ................................................................. 2
  na_if_in .................................................................. 2
Description
A dataset containing fake demographic data, used in the fauxnaif vignette.

Usage
faux_census

Format
A tibble with 20 rows and 6 variables.

Source
Fabricated

Description
This is a replacement for dplyr::na_if(). It is useful if you want to convert annoying values to NA. Unlike dplyr::na_if(), this function allows you to specify multiple values to be replaced with NA at the same time.

- na_if_in() replaces values that match its arguments with NA.
- na_if_not() replaces values that do not match its arguments with NA.

Usage
na_if_in(x, ...)
na_if_not(x, ...)

Arguments
x Vector to modify
... Values to replace with NA, specified as either:
  • An object, vector of objects, or list of objects.
  • A function (including a purrr-style lambda function) that returns a logical vector of the same length as x. See section "Formulas" for more details.
Value

A modified version of \( x \) with selected values replaced with \( \text{NA} \).

Formulas

These functions accept one-sided formulas that can evaluate to logical vectors of the same length as \( x \). The input is represented in these conditional statements as "\( . \)". Valid formulas take the form ~ \( . \) < 0. See examples.

See Also

dplyr::na_if() to replace a single value with \( \text{NA} \).
dplyr::coalesce() to replace missing values with a specified value.
tidyrr::replace_na() to replace \( \text{NA} \) with a value.
dplyr::recode() and dplyr::case_when() to more generally replace values.

Examples

```r
x <- sample(c(1:5, 99))
# We can replace 99...
# ... explicitly
na_if_in(x, 99)
# ... by specifying values to keep
na_if_not(x, 1:5)
# ... or by using a formula
na_if_in(x, ~ . > 5)

messy_string <- c("abc", ",", "def", "NA", "ghi", 42, "jkl", "NULL", "mno")
# We can replace unwanted values...
# ... one at a time
clean_string <- na_if_in(messy_string, ")")
clean_string <- na_if_in(clean_string, "NA")
clean_string <- na_if_in(clean_string, 42)
clean_string <- na_if_in(clean_string, "NULL")
clean_string
# ... or all at once
na_if_in(messy_string, ",", "NA", "NULL", 1:100)
na_if_in(messy_string, c(",", "NA", "NULL", 1:100))
na_if_in(messy_string, list(",", "NA", "NULL", 1:100))
# ... or using a clever formula
grepl("[a-z]{3,}\", messy_string)
na_if_not(messy_string, ~ grepl("[a-z]{3,}\", .))

# na_if_in() is particularly useful inside dplyr::mutate
library(dplyr)
faux_census %>%
  mutate(
    state = na_if_in(state, "Canada"),
    age = na_if_in(age, ~ . < 18, ~ . > 120)
  )
```
# This function handles vector values differently than dplyr,
# and returns a different result with vector replacement values:
na_if_in(1:5, 5:1)
dplyr::na_if(1:5, 5:1)
Index

* datasets
  faux_census, 2
  dplyr::case_when(), 3
  dplyr::coalesce(), 3
  dplyr::na_if(), 2, 3
  dplyr::recode(), 3

  faux_census, 2
  na_if_in, 2
  na_if_not(na_if_in), 2

  purrr-style lambda function, 2
  tidyr::replace_na(), 3