

# Package: fauxnaif (via r-universe)

September 9, 2024

**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

**URL** <https://fauxnaif.rossellhayes.com/>,  
<https://github.com/rossellhayes/fauxnaif>

**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

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.1

**Repository** <https://rossellhayes.r-universe.dev>

**RemoteUrl** <https://github.com/rossellhayes/fauxnaif>

**RemoteRef** HEAD

**RemoteSha** 9eb0d51409a1e291576f9d041b233a803f987056

## Contents

faux_census . . . . .	2
na_if_in . . . . .	2

<b>Index</b>	<b>5</b>
--------------	----------

---

faux_census	<i>A small sample of a fabricated census-like dataset</i>
-------------	---

---

**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

---

na_if_in	<i>Convert values to NA</i>
----------	-----------------------------

---

**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: <ul style="list-style-type: none"><li>• An object, vector of objects, or list of objects.</li><li>• A function (including a <a href="#">purrr-style lambda function</a>) that returns a logical vector of the same length as x. See section "Formulas" for more details.</li></ul> |

**Value**

A modified version of `x` with selected values replaced with 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 NA.

[dplyr::coalesce\(\)](#) to replace missing values with a specified value.

[tidyr::replace\\_na\(\)](#) to replace NA with a value.

[dplyr::recode\(\)](#) and [dplyr::case\\_when\(\)](#) to more generally replace values.

**Examples**

```
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