Package 'naflex'

October 15, 2024

Type Package

Title Flexible Options for Handling Missing Values

Version 0.1.1

Description For use in summary functions to omit missing values conditionally using specified checks.

License LGPL (>= 3)

URL https://github.com/dannyparsons/naflex

BugReports https://github.com/dannyparsons/naflex/issues

Imports stats

Suggests knitr, magrittr, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Danny Parsons [aut, cre] (<https://orcid.org/0000-0002-8333-9880>), Shadrack Kibet [aut], Lily Clements [ctb]

Maintainer Danny Parsons <danny@idems.international>

Repository CRAN

Date/Publication 2024-10-15 08:00:06 UTC

Contents

naflex *naflex: Flexible Options for Handling Missing Values*

Description

The naflex package provides additional flexibility for handling missing values in summary functions beyond the two extreme options (na.rm = TRUE/FALSE) available in base R.

Details

Most summary functions in R e.g. mean provide the option for the two extremes:

- calculate the summary ignoring all missing values, na.rm = TRUE, or
- require no missing values for the summary to be calculated, na.rm = FALSE

In many applications something in between these two extremes is often appropriate. For example, you may wish to give a summary statistic if less than 5% of values are missing.

naflex provides helper functions to facilitate this flexibility for dealing with missing values, particularly within summary functions.

In particular naflex provides four types of missing value checks:

- a maximum proportion of missing values allowed
- a maximum number of missing values allowed
- a maximum number of consecutive missing values allowed, and
- a minimum number of non-missing values required.

These checks can be used individually or in combination with each other within summary functions.

Author(s)

Maintainer: Danny Parsons <danny@idems.international> (ORCID)

Authors:

- Shadrack Kibet

See Also

Useful links:

- https://github.com/dannyparsons/naflex
- Report bugs at https://github.com/dannyparsons/naflex/issues

na_check *Check missing values conditions*

Description

na_check checks conditions on missing values in a vector. If all the checks pass it returns TRUE, otherwise FALSE.

Usage

```
na_check(
  x,
  prop = NULL,
  n = NULL,
  consec = NULL,
  n_non = NULL,
  prop_strict = FALSE
)
```

Arguments

x	Vector to check the missing values properties of.
prop	The maximum proportion (0 to 1) of missing values allowed.
n	The maximum number of missing values allowed.
consec	The maximum number of consecutive missing values allowed.
n_non	The minimum number of **non-missing** values required.
prop_strict	A logical (default FALSE) indicating if the proportion of missing values must be **strictly** less than prop (prop_strict = TRUE) or only less than prop (prop_strict = FALSE).

Details

There are four type of checks available:

- a maximum proportion of missing values allowed (prop)
- a maximum number of missing values allowed (n)
- a maximum number of consecutive missing values allowed (consec), and
- a minimum number of non-missing values required (n_non).

Any number of checks may be specified, including none. If multiple checks are specified, they must all pass in order to return TRUE. If no checks are specified then TRUE is returned, since this is considered as "all" checks passing.

Value

TRUE if all specified checks pass, and FALSE otherwise.

Examples

```
x <- c(1:3, NA, NA, NA, 4, NA, NA, 3)
# check if no more than 50% of values are missing
na_check(x, prop = 0.5)
# check if no more than 50% of values are missing
# and if there are no more than 2 consecutive missing values.
na_check(x, prop = 0.5, consec = 2)
```

`na_check_prop` *Check missing values conditions (single condition)*

Description

These set of functions check a condition on missing values in a vector `x`. They return `TRUE` if check passes, and `FALSE` otherwise. They are special cases of `na_check`, which is the general case for specifying multiple checks.

Usage

```
na_check_prop(x, prop = NULL, strict = FALSE)

na_check_n(x, n = NULL)

na_check_consec(x, consec = NULL)

na_check_non_na(x, n_non = NULL)
```

Arguments

`x`	Vector to check the missing values properties of.
`prop`	The maximum proportion (0 to 1) of missing values allowed.
`strict`	A logical (default `FALSE`) indicating if the proportion of missing values must be **strictly** less than `prop` (`strict = TRUE`) or only less than `prop` (`strict = FALSE`).
`n`	The maximum number of missing values allowed.
`consec`	The maximum number of consecutive missing values allowed.
`n_non`	The minimum number of **non-missing** values required.

Details

These functions replicate the functionality of `na_check` as individual functions for single checks.

For example, `na_check_n(x, 5)` is equivalent to `na_check(x, n = 5)`.

This more restricted form may be desirable when only a single check is required.

Value

These functions return TRUE if the check passes, and FALSE otherwise.

They are convenient wrapper functions for:

- na_prop(x) <= prop or na_prop(x) < prop (if strict = TRUE)
- na_n(x) <= n
- na_consec(x) <= consec
- na_non_na(x) >= n_non

Examples

```
na_check_prop(c(1, 2, NA, 4), 0.6)

na_check_prop(c(1, 2, NA, 4), 0.4)

na_check_prop(c(1:10, NA), 0.1)

na_check_prop(c(1:9, NA), 0.1, strict = TRUE)

na_check_n(c(1, 2, NA, 4, NA, NA, 7), 5)

na_check_n(c(1:9, NA, NA, NA), 2)

na_check_consec(c(1, NA, NA, NA, 2, NA, NA, 7), 4)

na_check_consec(c(rep(NA, 5), 1:2, rep(NA, 6)), 5)

na_check_non_na(c(1, 2, NA, 4, NA, NA, 7), 5)

na_check_non_na(c(1:9, NA, NA, NA), 2)
```

---

na_omit_if *Conditionally omit missing values*

Description

na_omit_if removes missing values from x if the specified checks are satisfied, and returns x unmodified otherwise. When used within summary functions, na_omit_if provides greater flexibility than the na.rm option e.g. sum(na_omit_if(x, prop = 0.05)).

Usage

```
na_omit_if(
  x,
  prop = NULL,
  n = NULL,
  consec = NULL,
```

```
  n_non = NULL,
  prop_strict = FALSE
)
```

Arguments

x	Vector to omit missing values in if checks pass.
prop	The maximum proportion (0 to 1) of missing values allowed.
n	The maximum number of missing values allowed.
consec	The maximum number of consecutive missing values allowed.
n_non	The minimum number of **non-missing** values required.
prop_strict	A logical (default FALSE) indicating if the proportion of missing values must be **strictly** less than prop (prop_strict = TRUE) or only less than prop (prop_strict = FALSE).

Details

There are four type of checks available:

- a maximum proportion of missing values allowed (prop)
- a maximum number of missing values allowed (n)
- a maximum number of consecutive missing values allowed (consec), and
- a minimum number of non-missing values required (n_non).

Any number of checks may be specified, including none. If multiple checks are specified, they must all pass in order for missing values to be omitted. If no checks are specified then missing values are omitted, since this is considered as "all" checks passing.

Value

A vector of the same type as x. Either x with missing values removed if all checks pass, or x unmodified if any checks fail.

For consistency with na.omit, if missing values are removed, the indices of the removed values form an na.action attribute of class omit in the result.

If missing values are not removed (because the checks failed or there were no missing values in x) then no na.action attribute is added.

Examples

```
x <- c(1, 3, NA, NA, NA, 4, 2, NA, 4, 6)

sum(na_omit_if(x, prop = 0.45, n = 10, consec = 5))
sum(na_omit_if(x, prop = 0.45))

require(magrittr)
sum(x %>% na_omit_if(prop = 0.45))

# WMO specification for calculating monthly values from daily data
```

```
daily_rain <- rnorm(30)
daily_rain[c(3, 5, 6, 7, 8, 9, 24, 28)] <- NA
sum(daily_rain %>% na_omit_if(n = 10, consec = 4))
```

`na_omit_if_prop` *Conditionally omit missing values (single condition)*

Description

These set of functions remove missing values from `x` if the single, specified check is satisfied, and returns `x` unmodified otherwise. They are special cases of `na_omit_if`, which is the general case for specifying multiple checks.

Usage

```
na_omit_if_prop(x, prop = NULL, strict = FALSE)

na_omit_if_n(x, n = NULL)

na_omit_if_consec(x, consec = NULL)

na_omit_if_non_na(x, n_non = NULL)
```

Arguments

`x`	Vector to omit missing values in if checks pass.
`prop`	The maximum proportion (0 to 1) of missing values allowed.
`strict`	A logical (default `FALSE`) indicating if the proportion of missing values must be **strictly** less than `prop` (`strict = TRUE`) or only less than `prop` (`strict = FALSE`).
`n`	The maximum number of missing values allowed.
`consec`	The maximum number of consecutive missing values allowed.
`n_non`	The minimum number of **non-missing** values required.

Details

These functions replicate the functionality of `na_omit_if` as individual functions for single checks.

For example, `na_omit_if_consec(x, 4)` is equivalent to `na_omit_if(x, consec = 4)`.

This more restricted form may be desirable when only a single check is required.

Value

A vector of the same type as `x`. Either `x` with missing values removed if all checks pass, or `x` unmodified if any checks fail.

For consistency with `na.omit`, if missing values are removed, the indices of the removed values form an `na.action` attribute of class `omit` in the result.

If missing values are not removed (because the checks failed or there were no missing values in `x`) then no `na.action` attribute is added.

`na_prop` *Calculate missing value properties*

Description

A set of functions for calculating missing values properties of a vector.

- `na_prop`: The proportion of missing values
- `na_n`: The number of missing values
- `na_consec`: The maximum number of consecutive missing values
- `na_non_na`: The number of non-missing values

Usage

```
na_prop(x)

na_n(x)

na_consec(x)

na_non_na(x)
```

Arguments

`x`	A vector to calculate the missing values property of.

Details

These functions are used by `na_omit_if` to omit missing values conditionally on the value of these properties. They are also useful summaries in their own right.

Value

Each function returns a number: a proportion (0 to 1) or a count.

Examples

```
na_prop(c(1, 2, NA, 4))

na_prop(c(1:9, NA))

na_n(c(1, 2, NA, 4, NA, NA, 7))

na_n(c(1:9, NA, NA))

na_consec(c(1, NA, NA, NA, 2, NA, NA, 7))

na_consec(c(rep(NA, 5), 1:2, rep(NA, 6)))

na_non_na(c(1, 2, NA, 4, NA, NA, 7))

na_non_na(c(1:9, NA, NA))
```

Index