Package: vek (via r-universe)

August 22, 2024

Title Predicate Helper Functions for Testing Simple Atomic Vectors

Version 1.0.0

Maintainer Sam Semegne <sam.ahoi@hotmail.com>

Description Predicate helper functions for testing atomic vectors in R. All functions take a single argument 'x' and check whether it's of the target type of base-R atomic vector (i.e. no class extensions nor attributes other than 'names'), returning TRUE or FALSE. Some additionally check for value (e.g. absence of missing values, infinities, blank characters, or 'names' attribute; or having length 1).

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Depends R (>= 3.0.0)

Suggests tinytest (>= 1.4.1)

URL https://github.com/samsemegne/vek

Repository https://samsemegne.r-universe.dev

RemoteUrl https://github.com/samsemegne/vek

RemoteRef HEAD

RemoteSha 939b023d8f940b2a46f924799a8808dbe7e46a75

Contents

vek *vek*

Description

Predicate helper functions for testing atomic vectors.

All functions take a single argument `x` and check whether it's of the target type of base-R atomic vector, returning `TRUE` or `FALSE`. Some additionally check for value. Classes that extend any base-R atomic vector return `FALSE`. Vectors that carry any attributes other than 'names' return `FALSE`.

Function names may include a suffix that encodes what additional conditions are evaluated. A select combination of these conditions is provided for each type. Naming scheme:

- n: no 'names' attribute
- x: no `NA` (type specific)
- y: no `NaN`
- z: no `Inf` or `-Inf`
- b: no blank characters, i.e. `""`
- 1: is of length 1

For example:

- `is_dbl_vec(x)` evaluates whether `x` is a base-R typeof double atomic vector, of any length.
- `is_dbl_vec_xz(x)` will additionaly evaluate whether `x` contains no `NA` nor `Inf` values. Note, `NaN` values are still allowed here, which is dissimilar to behavior from base-R functions like `is.na(x)` or `anyNA(x)`, wherein both `NA` and `NaN` values yield `TRUE`.
- `is_num_vec_xyz1(x)` effectively evaluates whether `x` is a single real number.

Supported types: logical, integer, double, numeric, and character.

Usage

```
is_lgl_vec(x)

is_lgl_vec_x(x)

is_lgl_vec_x1(x)

is_lgl_vec_n(x)

is_lgl_vec_nx(x)

is_lgl_vec_nx1(x)

is_int_vec(x)
```

```
is_int_vec_x(x)

is_int_vec_x1(x)

is_int_vec_n(x)

is_int_vec_nx(x)

is_int_vec_nx1(x)

is_dbl_vec(x)

is_dbl_vec_x(x)

is_dbl_vec_y(x)

is_dbl_vec_z(x)

is_dbl_vec_xy(x)

is_dbl_vec_yz(x)

is_dbl_vec_xyz(x)

is_dbl_vec_x1(x)

is_dbl_vec_y1(x)

is_dbl_vec_xy1(x)

is_dbl_vec_yz1(x)

is_dbl_vec_xyz1(x)

is_dbl_vec_n(x)

is_dbl_vec_nx(x)

is_dbl_vec_ny(x)

is_dbl_vec_nz(x)

is_dbl_vec_nxy(x)

is_dbl_vec_nyz(x)

is_dbl_vec_nxyz(x)
```

```
is_dbl_vec_nx1(x)

is_dbl_vec_ny1(x)

is_dbl_vec_nxy1(x)

is_dbl_vec_nyz1(x)

is_dbl_vec_nxyz1(x)

is_chr_vec(x)

is_chr_vec_x(x)

is_chr_vec_b(x)

is_chr_vec_xb(x)

is_chr_vec_x1(x)

is_chr_vec_b1(x)

is_chr_vec_xb1(x)

is_chr_vec_n(x)

is_chr_vec_nx(x)

is_chr_vec_nb(x)

is_chr_vec_nxb(x)

is_chr_vec_nx1(x)

is_chr_vec_nb1(x)

is_chr_vec_nxb1(x)

is_num_vec(x)

is_num_vec_x(x)

is_num_vec_y(x)

is_num_vec_z(x)

is_num_vec_xy(x)
```

```
is_num_vec_yz(x)

is_num_vec_xyz(x)

is_num_vec_x1(x)

is_num_vec_y1(x)

is_num_vec_yz1(x)

is_num_vec_xy1(x)

is_num_vec_xyz1(x)

is_num_vec_n(x)

is_num_vec_nx(x)

is_num_vec_ny(x)

is_num_vec_nz(x)

is_num_vec_nxy(x)

is_num_vec_nyz(x)

is_num_vec_nxyz(x)

is_num_vec_nx1(x)

is_num_vec_ny1(x)

is_num_vec_nyz1(x)

is_num_vec_nxy1(x)

is_num_vec_nxyz1(x)
```

Arguments

x	any.

Value

`logical`. Returns `TRUE` or `FALSE`.

See Also

`vector()`

Index