Package: isstatic (via r-universe)

August 26, 2024

Type Package

Title Dependency-Free Object Tests

Version 0.1.0.9000

Maintainer Eli Pousson <eli.pousson@gmail.com>

Description Convenience functions for checking class inheritance, extracting attributes, basic type conversion, and miscellaneous string manipulation. working with sf, ggplot2, and other packages.

License CC0

URL https://github.com/elipousson/isstatic, https://elipousson.github.io/isstatic/

BugReports https://github.com/elipousson/isstatic/issues

Suggests testthat (>= 3.0.0)

Remotes wch/staticimports

Config/Needs/roxygen wch/staticimports

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

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

RemoteUrl https://github.com/elipousson/isstatic

RemoteRef HEAD

RemoteSha 29ce2ff9a0206386c67cc41585f11fa8be42f947

Contents

`alpha_to_int` *Convert an alphabetical character object from A to Z into a corresponding integer*

Description

Integers and NA values are passed as is. Double or characters with no corresponding Roman numeral are converting to NA values.

Usage

```
alpha_to_int(x, dict = LETTERS, n = 1, quiet = TRUE, call = parent.frame())
```

Arguments

`x`	Character vector of length n strings to compare to dict. Typically, letters from "A" to "Z". Case sensitive.
`dict`	Character vector to match to x. Default: LETTERS.
`n`	Maximum character length for non-NA objects permitted. Set to NULL or >1 if dict includes objects with more than one character.
`quiet`	If `TRUE`, suppress warnings for introduction of NA values through coercion.
`call`	Default: `parent.frame()`. Passed to input checking functions to improve error messages.

Value

A length 1 integer between 1 and 26.

`as_cardinal_bearing` *Convert a numeric bearing value to the closest cardinal bearing*

Description

Convert a numeric bearing value to the closest cardinal bearing

Usage

```
as_cardinal_bearing(x, winds = 8, cols = c("bearing", "cardinal_bearing"))
```

Arguments

x	A numeric vector with degrees or a data.frame with column name matching the first name in cols.
winds	Number of winds to use for results (4, 8, or 16).
cols	A length 2 character vector where the first value is a column name containing bearing values and the second is the name of the new column added to the data.frame. Required if x is a data.frame.

Value

A named numeric vector with cardinal bearings (and wind names) or a data.frame with an added column containing the cardinal bearings.

---

as_crs *Coerce a sf, sfc, or bbox object to a coordinate reference system*

Description

This function should be updated to support stars and terra objects.

Usage

```
as_crs(x, input = TRUE)

is_lonlat_crs(x, crs = c("EPSG:4326", "EPSG:4269"))
```

Arguments

x	A sf, sfc, or bbox object to coerce into a CRS.
input	If TRUE (default), return only the "input" component of the crs object. If FALSE, return the full crs object.
crs	For is_lonlat_crs, coordinate reference system to use as lonlat crs.

---

as_integer *as.integer with option to suppress warnings for NA coercion*

Description

as.integer with option to suppress warnings for NA coercion

Usage

```
as_integer(x, quiet = TRUE)
```

Arguments

x	object to be coerced or tested.
quiet	If TRUE, suppress warnings about creation of NA values through coercion of object types. Default to TRUE.

---

as_numbered_labels *Convert a numeric vector to a vector of numbered labels*

Description

This function allows the creation of numbered labels for a vector using a range of numbering styles.

Usage

```
as_numbered_labels(
  x,
  labels = "arabic",
  start = NULL,
  suffix = NULL,
  base = 26,
  cols = "num_label",
  pad = NULL,
  side = "left",
  quiet = TRUE,
  call = parent.frame()
)
```

Arguments

x	An integer or other vector or a data.frame. An integer vector or integer column is used as the number that is converted based on the label style. If x is not an integer or data.frame with an integer column, the numbering is created based on seq_along().
labels	Label style. Options include "arabic", "alph", "Alph", "alpha", "Alpha", "roman", or "Roman".
start	Starting number or value. Letters are supported if label style is "alph", "Alph", "alpha", or "Alpha" and Roman numerals are supported if label is "roman" or "Roman".
suffix	Suffix character to follow number labels. For example, if x = 1 and suffix = ". " the returned label would be "1."
base	Base used in alphabetical number labels. Highest letter to use for alphabetical numbers. Single digit letters (A to Z) or numbers 1 to 26 are supported. For example, if base is 3, alphabetical labels for numbers higher than 3 have the prior value prefixed so 3 would be "C" and 4 would be "AA". Defaults to 26 which converts 27 to "AA", 53 to "BA", etc.

cols	Column name to use for added column for number labels when x is a data.frame. Defaults to "num_label". If cols is length 2, the first item in the vector is assumed to be the column name from the data.frame to use as x and the second item is used as the column name for the added column with number labels.
pad, side	If pad is not NULL, pass pad and side to str_pad() added from the stringstatic::str_pad() package.
quiet	If TRUE, suppress warnings about creation of NA values through coercion of object types. Default to TRUE.
call	Default: parent.frame(). Passed to input checking functions to improve error message traceback.

Value

- If x is a vector, function returns numeric vector if labels is "arabic" or a character vector otherwise.
- If x is a data.frame, as_numbered_labels() returns a modified data.frame with an added column with a name matching the second value of the cols parameter.

---

as_orientation *What is the orientation of a numeric aspect ratio?*

Description

What is the orientation of a numeric aspect ratio?

Usage

```
as_orientation(x, tolerance = 0.1, cols = c("width", "height"))
```

Arguments

x	A numeric vector with an aspect ratio or a data.frame with width and height column (using width and height values from columns matching the cols parameter).
tolerance	Positive numeric value above or below 1 used to determine if an aspect ratio is square, landscape, or portrait.
cols	Name of width and height column if x is a data.frame object.

Value

A character vector of orientations of the same length as x or, if x is a data.frame, the same length as the number of rows in x.

as_roman *as.roman with option to suppress warnings for NA coercion*

Description

as.roman with option to suppress warnings for NA coercion

Usage

```
as_roman(x, quiet = TRUE)
```

Arguments

x	a numeric or character vector of arabic or roman numerals.
quiet	If TRUE, suppress warnings about creation of NA values through coercion of object types. Default to TRUE.

as_sfc *Coerce a sf object to a sfc object*

Description

If possible, function should be updated to support bbox objects or other spatial data classes.

Usage

```
as_sfc(x)
```

Arguments

x	A sf object to coerce.

`combine_words` *Combine multiple words into a single string*

Description

When a value from an inline R expression is a character vector of multiple elements, we may want to combine them into a phrase like 'a and b', or a, b, and c. That is what this a helper function does.

Usage

```
combine_words(
  words,
  sep = ", ",
  and = " and ",
  before = "",
  after = before,
  oxford_comma = TRUE
)
```

Arguments

words	A character vector.
sep	Separator to be inserted between words.
and	Character string to be prepended to the last word.
before, after	A character string to be added before/after each word.
oxford_comma	Whether to insert the separator between the last two elements in the list.

Details

If the length of the input words is smaller than or equal to 1, words is returned. When words is of length 2, the first word and second word are combined using the and string, or if blank, sep if is used. When the length is greater than 2, sep is used to separate all words, and the and string is prepended to the last word.

Value

A character string

Author(s)

Yihui Xie <xie@yihui.name> (ORCID)

Source

Adapted from knitr::combine_words() in the knitr package.

Examples

```
combine_words("a")
combine_words(c("a", "b"))
combine_words(c("a", "b", "c"))
combine_words(c("a", "b", "c"), sep = " / ", and = "")
combine_words(c("a", "b", "c"), and = "")
combine_words(c("a", "b", "c"), before = "\"", after = "\"")
combine_words(c("a", "b", "c"), before = "\"", after = "\"", oxford_comma = FALSE)
```

---

digit_pattern *Helper to return a regex based on side*

Description

Helper to return a regex based on side

Usage

```
digit_pattern(pattern = "[0-9]+", side = NULL)
```

---

file_path *Construct path to file ignoring NULL values for filename or path*

Description

A replacement for `file.path()`

Usage

```
file_path(
  ...,
  path = NULL,
  filename = NULL,
  fsep = .Platform$file.sep,
  allow_null = FALSE,
  call = parent.frame()
)
```

Arguments

`...`	Additional strings to pass before path and filename.
`path`	Path name. Optional if filename is supplied.
`filename`	File name. Optional if path is supplied.
`fsep`	the path separator to use (assumed to be ASCII).
`allow_null`	If `TRUE`, return `NULL` if filename and path are `NULL` and no additional strings are provided to .... If `FALSE`, stop if filename and path are `NULL` and no additional strings are provided to ...

has_fileext *Does string contain the specified file type or any file extension?*

Description

Check if string contains any filetype or the provided filetype. If string is NULL, returns FALSE.

Usage

```
has_fileext(string = NULL, fileext = NULL, ignore.case = FALSE)
```

Arguments

string	String to be tested with or without filetype. Defaults to NULL.
fileext	File type to test against. Optional.
ignore.case	If FALSE, the pattern matching is case sensitive. If TRUE, case is ignored.

See Also

is_fileext_path()

has_filetype *Does string contain the specified file type or any file extension?*

Description

Alternate naming convention for has_fileext()

Usage

```
has_filetype(string = NULL, filetype = NULL, ignore.case = FALSE)
```

Arguments

string	String to be tested with or without filetype. Defaults to NULL.
filetype	File type to test against. Optional.
ignore.case	If FALSE, the pattern matching is case sensitive. If TRUE, case is ignored.

has_len_between *Is the length of x between two values?*

Description

Is the length of x between two values?

Usage

```
has_len_between(x, left = 1, right = left)

has_min_length(x, min)

has_max_length(x, max)
```

Arguments

x	Object to check.
left, right	Min and max values to check if the length of x is between.
min	Min value used by has_min_length().
max	Max value used by has_max_length().

has_same_crs

- has_same_crs(): *Do two sf, sfc, or bbox objects use the same coordinate reference system?*

Description

- has_same_crs(): Do two sf, sfc, or bbox objects use the same coordinate reference system?

Usage

```
has_same_crs(x, y, ...)
```

Arguments

x, y	sf, sfc, or bbox objects to be compared.
...	Additional parameters passed to identical() by has_same_crs().

has_same_len *Do two object have an identical length?*

Description

Do two object have an identical length?

Do two object have an identical length?

Usage

```
has_same_len(x, y, ...)
```


```
has_same_len(x, y, ...)
```


Arguments

x, y	Two strings or character vectors to compare.
...	Additional parameters passed to `identical()`

int_to_alpha *Convert a integer into a corresponding letter or multi-letter string*

Description

Character values in the provided dict (default to letters "A" to "Z") are passed as is. Non-integer numeric values or characters that are not found in the provided dict are converting to NA values.

Usage

```
int_to_alpha(x, suffix = NULL, base = 26, dict = LETTERS, quiet = TRUE)
```

Arguments

x	An integer vector or a vector that can be coerced to an integer vector
suffix	Suffix character to follow alpha character, e.g. if `x = 1` and `suffix = "."` the returned label would be "A.". suffix is also used to separate values when x is greater than base, e.g. `x = 27` and `suffix = "."` returns "A.A." Defaults to `NULL`.
base	If base is not numeric, it is converted to an integer with `alpha_to_int()`.
dict	Character vector to compare to x. Default: LETTERS.
quiet	If `TRUE`, suppress warnings for introduction of NA values through coercion.

Value

An integer vector composed of objects between 1 and 26 with the same length as x.

Source

Adapted from the recursive solution provided by G. Grothendieck in a May 31, 2017 StackOverflow answer.

`is_all` *Do all items in a list or vector return TRUE from a predicate function?*

Description

Do all items in a list or vector return TRUE from a predicate function?

- `is_all_null()`: Are all items in a list or vector NULL values?
- `is_all_na()`: Are all items in a list or vector NA values?

Usage

```
is_all(x, FUN, ...)

is_all_null(x)

is_all_na(x)
```

Arguments

x	A list or vector passed to X parameter of `vapply()`.
FUN	the function to be applied to each element of X: see 'Details'. In the case of functions like +, %*%, the function name must be backquoted or quoted.
...	Arguments passed on to `base::vapply`
	USE.NAMES logical; if TRUE and if X is character, use X as `names` for the result unless it had names already. Since this argument follows ... its name cannot be abbreviated.
	FUN.VALUE a (generalized) vector; a template for the return value from FUN. See 'Details'.

Value

TRUE if FUN returns TRUE for all elements of x or FALSE if any element returns FALSE.

See Also

`is_any()`

is_all_in *Is all of x in y?*

Description

Is all of x in y?

Usage

```
is_all_in(x, y)
```

Arguments

x	Object to be tested.
y	Vector to compare x to.

is_any *Do any items in a list or vector return* TRUE *from a predicate function?*

Description

Do any items in a list or vector return TRUE from a predicate function?

- is_any_null(): Is any item in a list or vector a NULL value?
- is_any_na(): Is any item in a list or vector a NA value?
- is_none(): Is no item in a list or vector return TRUE from a predicate function?
- is_none_null(): Is no item in a list or vector is NULL?

Do any items in a list or vector return TRUE from a predicate function?

- is_any_null(): Is any item in a list or vector a NULL value?
- is_any_na(): Is any item in a list or vector a NA value?

Usage

```
is_any(x, FUN, ...)

is_any_null(x)

is_any_na(x)

is_none(x, FUN, ...)

is_none_null(x)

is_any(x, FUN, ...)

is_any_null(x)

is_any_na(x)
```

Arguments

x	A list or vector passed to vapply().
FUN	the function to be applied to each element of X: see 'Details'. In the case of functions like +, %*%, the function name must be backquoted or quoted.
...	Arguments passed on to base::vapply, base::vapply
	USE.NAMES logical; if TRUE and if X is character, use X as names for the result unless it had names already. Since this argument follows ... its name cannot be abbreviated.
	FUN.VALUE a (generalized) vector; a template for the return value from FUN. See 'Details'.

Value

TRUE if FUN returns TRUE for any element of x or FALSE if all elements return FALSE.

TRUE if FUN returns TRUE for any element of x or FALSE if all elements return FALSE.

See Also

is_all()

is_all()

---

is_any_in *Are any of x in y?*

Description

Are any of x in y?

Usage

```
is_any_in(x, y)
```

Arguments

x	Object to be tested.
y	Vector to compare x to.

---

is_blank *Test if a character vector consists of blank strings*

Description

Return a logical vector indicating if elements of a character vector are blank (white spaces or empty strings).

Usage

```
is_blank(x)
```

Arguments

x	A character vector.

Value

TRUE for blank elements, or FALSE otherwise.

Author(s)

Yihui Xie <xie@yihui.name> (ORCID)

Source

Adapted from `xfun::is_blank()` in the xfun package.

Examples

```
is_blank("")
is_blank("abc")
is_blank(c("", "  ", "\n\t"))
is_blank(c("", " ", "abc"))
```

is_file *Is x a file or directory?*

Description

`is_file()` is a wrapper for `base::file.exists()` that allows the exclusion of directories and returning named vectors. `is_dir()` is a wrapper for `base::dir.exists()` that supports vector inputs rather than single strings. character(0) inputs return `FALSE`.

Usage

```
is_file(x, include_dirs = FALSE, use_names = FALSE)

is_dir(x, use_names = FALSE)
```

Arguments

`include_dirs`	If `TRUE`, return `TRUE` for any value of x that is a directory path. If `FALSE` (default), return `FALSE` for directory paths.
`use_names`	If `TRUE`, return a logical vector where the names match the values of the input vector x. Defaults to `FALSE`.

is_fileext_path *Does this text end in the provided file extension?*

Description

Does this text end in the provided file extension?

is_geojson_fileext: Does this text end with a GeoJSON file extension?

is_csv_fileext: Does this text end with a CSV file extension?

is_excel_fileext: Does this text end with a XLS or XLSX file extension?

is_rdata_fileext: Does this text end with a rds, rda, or RData file extension?

is_rds_fileext: Does this text end with a rds file extension?

is_rda_fileext: Does this text end with a rda file extension?

is_zip_fileext: Does this text end with a zip file extension?

Usage

```
is_fileext_path(x, fileext, ignore.case = TRUE)

is_geojson_fileext(x, ignore.case = TRUE)

is_csv_fileext(x, ignore.case = TRUE)

is_excel_fileext(x, ignore.case = TRUE)

is_rdata_fileext(x, ignore.case = TRUE)

is_rds_fileext(x, ignore.case = TRUE)

is_rda_fileext(x, ignore.case = TRUE)

is_zip_fileext(x, ignore.case = TRUE)
```

Arguments

x	A character vector to check for matches, or an object which can be coerced by as.character() to a character vector.
fileext	A file extension to compare to x. Required. If a vector of multiple extensions are provided, returns TRUE for any match.
ignore.case	if FALSE, the pattern matching is *case sensitive* and if TRUE, case is ignored during matching.

See Also

has_fileext()

---

is_filetype_path *Does this text end in the specified filetype?*

Description

Does this text end in the specified filetype?

Usage

```
is_filetype_path(x, filetype, ignore.case = TRUE)
```

Arguments

x	A character vector to check for matches, or an object which can be coerced by as.character() to a character vector.
filetype	A file extension (or multiple file extensions) to compare to x. Required.

ignore.case — if FALSE, the pattern matching is *case sensitive* and if TRUE, case is ignored during matching.

---

is_gg — *Is this a gg class object?*

Description

Is this a gg class object?

Is this a ggplot class object?

Is this a ggproto class object?

Is this a patchwork class object?

Usage

```
is_gg(x)

is_ggplot(x)

is_ggproto(x)

is_patchwork(x)
```

Arguments

x — Object to be tested.

---

is_gg_list — *Do all items in this list inherit the gg class?*

Description

Do all items in this list inherit the gg class?

Usage

```
is_gg_list(x)
```

Arguments

x — Object to be tested.

is_list_all *Do all items in this list inherit the provided class?*

Description

Do all items in this list inherit the provided class?

Usage

```
is_list_all(x, what = NULL)
```

Arguments

x	Object to be tested.
what	A character vector naming classes.

is_named *Is this object a named list or character vector?*

Description

Is this object a named list or character vector?

Does this object have all of the provided names?

Does this object have any of the provided names?

Usage

```
is_named(x)

has_all_names(x, name)

has_any_names(x, name)
```

Arguments

x	A data frame or another named object.
name	Element name(s) to check.

See Also

rlang::is_named()

is_none_in *Are none of x in y?*

Description

Are none of x in y?

Usage

```
is_none_in(x, y)
```

Arguments

x	Object to be tested.
y	Vector to compare x to.

is_sf *Is this a sf class object?*

Description

Is this a sf class object?

Is this a sfc class object?

Is this a sfg class object?

Is this a bbox class object?

Is this a sf, sfc, or bbox class object?

Is this a RasterLayer class object?

Is this a Spatial class (sp) object?

Usage

```
is_sf(x)

is_sfc(x)

is_sfg(x)

is_bbox(x)

is_sf_ext(x, ext = TRUE)

is_raster(x)

is_sp(x)
```

Arguments

x	An object to be tested with inherits()
ext	If TRUE, return TRUE is x is a sf, sfc, or bbox object. If FALSE, only check if x is an sf object. If ext is a character object, it is passed to the what parameter of inherits() with sf.

See Also

as_crs()

---

is_sf_list *Do all items in this list inherit the sf class?*

Description

Do all items in this list inherit the sf class?

Usage

```
is_sf_list(x)
```

Arguments

x	Object to be tested.

---

is_unit *Is this a unit class object?*

Description

Is this a unit class object?

Is this a margin class object?

Usage

```
is_unit(x)

is_margin(x)
```

Arguments

x	Object to be tested.

is_units *Is this a units class object?*

Description

Is this a units class object?

Usage

```
is_units(x)
```

Arguments

x	Object to be tested.

is_url *Is an object a URL?*

Description

Is an object a URL?

- is_esri_url(): Is an object an ArcGIS MapServer or FeatureServer URL?
- is_gsheet_url(): Is an object a Google Sheets URL?
- is_gist_url(): Is an object a URL for a GitHub Gist?
- is_gmap_url(): Is an object a Google Maps URL?

Usage

```
is_url(x)

is_esri_url(x)

is_gsheet_url(x)

is_gist_url(x)

is_gmap_url(x)
```

Arguments

x	A object to be tested.

map_chr *Apply a function to each element of a vector.*

Description

Apply a function to each element of a vector.

Usage

```
map_chr(.x, .f, ...)
```

Author(s)

Winston Chang <winston@stdout.org>

Source

purr-like functions in staticimports package

plural_words *Simple helper for pluralizing words*

Description

Simple helper for pluralizing words

Usage

```
plural_words(
  words,
  n = 1,
  suffix = "s",
  before = "",
  after = "",
  replacement = NULL
)
```

roman_to_int — *Convert a Roman numeral character object into a corresponding integer*

Description

Integers and NA objects are passed as is. Double numeric objects or characters with no corresponding Roman numeral are converting to NA values.

Usage

```
roman_to_int(x, quiet = TRUE)
```

Arguments

x	An integer vector or a character vector with characters representing Roman numerals.
quiet	If TRUE, suppress warnings for introduction of NA values through coercion.

set_start_number — *Set start number for numeric vector x*

Description

Helper for as_numbered_labels().

Usage

```
set_start_number(x, start = NULL, labels = "arabic")
```

Arguments

x	An integer or other vector or a data.frame. An integer vector or integer column is used as the number that is converted based on the label style. If x is not an integer or data.frame with an integer column, the numbering is created based on seq_along().
start	Starting number or value. Letters are supported if label style is "alph", "Alph", "alpha", or "Alpha" and Roman numerals are supported if label is "roman" or "Roman".
labels	Label style. Options include "arabic", "alph", "Alph", "alpha", "Alpha", "roman", or "Roman".

str_c *Join multiple strings into a single string*

Description

Dependency-free drop-in alternative for `stringr::str_c()`.

Usage

```
str_c(..., sep = "", collapse = NULL)
```

Arguments

`...`	One or more character vectors. Zero length arguments are removed. Short arguments are recycled to the length of the longest. Like most other R functions, missing values are "infectious": whenever a missing value is combined with another string the result will always be missing. Use `str_replace_na()` to convert `NA` to "NA"
`sep`	String to insert between input vectors.
`collapse`	Optional string used to combine input vectors into single string.

Value

If `collapse = NULL` (the default) a character vector with length equal to the longest input string. If collapse is non-`NULL`, a character vector of length 1.

Source

Adapted from the stringr package.

str_detect *Detect the presence or absence of a pattern in a string*

Description

Dependency-free drop-in alternative for `stringr::str_detect()`.

Usage

```
str_detect(string, pattern, negate = FALSE)
```

Arguments

`string`	Input vector. Either a character vector, or something coercible to one.
`pattern`	Pattern to look for. The default interpretation is a regular expression, as described in base::regex. Control options with `regex()`. Match a fixed string (i.e. by comparing only bytes), using `fixed()`. This is fast, but approximate.
`negate`	If `TRUE`, return non-matching elements.

Value

A logical vector.

Source

Adapted from the stringr package.

`str_extract` *Extract matching patterns from a string*

Description

Dependency-free drop-in alternative for `stringr::str_extract()`.

Usage

```
str_extract(string, pattern)
```

Arguments

`string`	Input vector. Either a character vector, or something coercible to one.
`pattern`	Pattern to look for. The default interpretation is a regular expression, as described in base::regex. Control options with `regex()`. Match a fixed string (i.e. by comparing only bytes), using `fixed()`. This is fast, but approximate.

Value

A character matrix. The first column is the complete match, followed by one column for each capture group.

Source

Adapted from the stringr package.

`str_fileext` *Add, remove, or extract file extensions from character vectors*

Description

These function uses `stringstatic::str_c()`, `stringstatic::str_remove()` and `stringstatic::str_extract()` and works to:

- Add file extensions (or replace existing file extensions) with `str_add_fileext()`
- Remove file extensions with `str_remove_fileext()`
- Extract existing file names `str_extract_fileext()` (returning NA values if a string has no file extension)

Usage

```
str_add_fileext(string, fileext = NULL)

str_remove_fileext(string, fileext = NULL)

str_extract_fileext(string, fileext = NULL)
```

Arguments

`string`	Character vector of any length. Required.
`fileext`	File extension. Optional. Defaults to `NULL`.

See Also

- `has_fileext()`
- `is_fileext_path()`

Examples

```
str_add_fileext("image", "jpeg")

str_remove_fileext(c("file.txt", "word.docx"), "docx")

str_extract_fileext(c("file1.pdf", "file2"))

str_extract_fileext(c("image1.png", "image2.jpeg"), "jpeg")
```

str_length *Compute the length of a string*

Description

Dependency-free drop-in alternative for `stringr::str_length()`.

Usage

```
str_length(string)
```

Arguments

`string`	Input vector. Either a character vector, or something coercible to one.

Value

A numeric vector the same length as string.

Source

Adapted from the stringr package.

str_n_freq *Get the n most frequent or least frequent appearing values in a vector*

Description

This function does not address ties in frequency.

Usage

```
str_n_freq(string = NULL, n = NULL, decreasing = TRUE)
```

Arguments

`string`	A character vector.
`n`	The number of values to return based on frequency of appearance. Defaults to `NULL` which returns all unique values from x.
`decreasing`	Passed to `sort()`

str_pad *Duplicate and concatenate strings within a character vector*

Description

Dependency-free drop-in alternative for `stringr::str_pad()`.

Usage

```
str_pad(
  string,
  width,
  side = c("left", "right", "both"),
  pad = " ",
  use_width = TRUE
)
```

Arguments

`string`	Input vector. Either a character vector, or something coercible to one.
`width`	Minimum width of padded strings.
`side`	Side on which padding character is added (left, right or both).
`pad`	Single padding character (default is a space).
`use_width`	If `FALSE`, use the length of the string instead of the width; see `str_width()`/`str_length()` for the difference.

Value

A character vector.

Author(s)

Eli Pousson <eli.pousson@gmail.com> (ORCID)

Alexander Rossell Hayes <alexander@rossellhayes.com> (ORCID)

Source

Adapted from the stringr package.

`str_pad_digits` *Modify digits within strings*

Description

`str_replace_digits()`: Replace digits with a string optionally incrementing the digits `str_pad_digits()`: Pad a string with digits `str_extract_digits()`: Extract digits from a string

Usage

```
str_pad_digits(string, pad = "0", side = "left", width = NULL)

str_extract_digits(string, pattern = "[0-9]+", side = NULL)

str_replace_digits(string, replacement, pad = "0", side = "left", width = NULL)

str_increment_digits(string, increment = TRUE, ...)
```

Arguments

`string`	Input vector. Either a character vector, or something coercible to one.
`pad`	Single padding character added to digits in string; defaults to "0"
`side`	Side on which padding character is added (left, right or both).
`width`	Minimum width of padded strings.
`pattern`	Pattern to look for. The default interpretation is a regular expression, as described in base::regex. Control options with `regex()`. Match a fixed string (i.e. by comparing only bytes), using `fixed()`. This is fast, but approximate.
`replacement`	A character vector of replacements. Should be either length one, or the same length as `string` or `pattern`. References of the form `\1`, `\2`, etc. will be replaced with the contents of the respective matched group (created by `()`). To replace the complete string with `NA`, use `replacement = NA_character_`. Using a function for `replacement` is not yet supported.
`increment`	If `TRUE`, increment digits in string by 1. If numeric, increment digits in string by value. If `NULL`, 0, or if no digits are present in string, return string as is.
`...`	Passed to `str_replace_digits()`

str_remove *Remove matched patterns in a string*

Description

Dependency-free drop-in alternative for `stringr::str_remove()`.

Usage

```
str_remove(string, pattern)
```

Arguments

`string`	Input vector. Either a character vector, or something coercible to one.
`pattern`	Pattern to look for. The default interpretation is a regular expression, as described in base::regex. Control options with `regex()`. Match a fixed string (i.e. by comparing only bytes), using `fixed()`. This is fast, but approximate.

Value

A character vector.

Source

Adapted from the stringr package.

str_replace *Replace matched patterns in a string*

Description

Dependency-free drop-in alternative for `stringr::str_replace()`.

Usage

```
str_replace(string, pattern, replacement)
```

Arguments

string	Input vector. Either a character vector, or something coercible to one.
pattern	Pattern to look for. The default interpretation is a regular expression, as described in base::regex. Control options with regex(). Match a fixed string (i.e. by comparing only bytes), using fixed(). This is fast, but approximate.
replacement	A character vector of replacements. Should be either length one, or the same length as string or pattern. References of the form \1, \2, etc. will be replaced with the contents of the respective matched group (created by ()). To replace the complete string with NA, use replacement = NA_character_. Using a function for replacement is not yet supported.

Value

A character vector.

Source

Adapted from the stringr package.

---

str_replace_all *Replace matched patterns in a string*

Description

Dependency-free drop-in alternative for stringr::str_replace_all().

Usage

```
str_replace_all(string, pattern, replacement)
```

Arguments

string	Input vector. Either a character vector, or something coercible to one.
pattern	Pattern to look for. The default interpretation is a regular expression, as described in base::regex. Control options with regex(). Match a fixed string (i.e. by comparing only bytes), using fixed(). This is fast, but approximate.
replacement	A character vector of replacements. Should be either length one, or the same length as string or pattern. References of the form \1, \2, etc. will be replaced with the contents of the respective matched group (created by ()). To perform multiple replacements in each element of string, pass a named vector (c(pattern1 = replacement1)) to str_replace_all(). To replace the complete string with NA, use replacement = NA_character_. Using a function for replacement is not yet supported.

Value

A character vector.

Source

Adapted from the stringr package.

---

str_width *Compute the width of a string*

Description

Dependency-free drop-in alternative for `stringr::str_width()`. Results for non-ASCII characters may be inaccurate in R < 4.0.

Usage

```
str_width(string)
```

Arguments

`string`	Input vector. Either a character vector, or something coercible to one.

Value

A numeric vector the same length as string.

Source

Adapted from the stringr package.

---

tosentence *Convert to a common sentence case*

Description

Convert to a common sentence case

Usage

```
tosentence(x)
```

Arguments

`x`	a character vector, or an object that can be coerced to character by `as.character()`.

Author(s)

Joachim Schork <info@joachimschork.com>

Source

Statistics Globe

`underscore` *Replace spaces in string with underscores*

Description

Replace spaces in string with underscores

Usage

```
underscore(string)
```

Arguments

`string`	String to transform by replacing spaces with underscores.

Index