Package: journalR (via r-universe)

November 10, 2026

Type Package

Title Formatting Tools for Scientific Journal Writing

Version 0.7.0

Maintainer Sam Byrne <ssbyrne@uw.edu>

Description Scientific journal numeric formatting policies implemented in code. Emphasis on formatting mean/upper/lower sets of values to pasteable text for journal submission. For example c(2e6, 1e6, 3e6) becomes ``2.00 million (1.00--3.00)''. Lancet and Nature have built-in styles for rounding and punctuation marks. Users may extend journal styles arbitrarily. Four metrics are supported; proportions, percentage points, counts and rates. Magnitudes for all metrics are discovered automatically.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.2.0)

Imports checkmate, data.table, glue

Suggests devtools (>= 2.4.5), testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.3

URL https://github.com/epi-sam/journalR

BugReports https://github.com/epi-sam/journalR/issues

Repository https://epi-sam.r-universe.dev

Date/Publication 2026-03-06 20:38:01 UTC

RemoteUrl https://github.com/epi-sam/journalr

RemoteRef HEAD

RemoteSha 57b1a68fc28cc4d01aaeabc237bb747fde646dfb

Contents

en_dash . . . 2
fmt_magnitude . . . 3
format_journal_clu . . . 4
format_journal_df . . . 6
format_lancet_clu . . . 7
format_lancet_df . . . 8
format_metric_cols . . . 10
format_nature_clu . . . 11
format_nature_df . . . 12
format_oxford_comma . . . 13
fround . . . 14
fround_metric . . . 15
fround_metric_lancet . . . 15
get_metric_labels . . . 16
get_metrics . . . 17
get_style . . . 18
get_style_schema . . . 18
mid_dot . . . 19
new_style . . . 19
set_magnitude . . . 22
set_style . . . 23
style_lancet . . . 24
style_nature . . . 25
thin_space . . . 25

Index **27**

`en_dash` *En-dash*

Description

Standard for "x – y" and Lancet negative: ("\U2013")

Usage

```
en_dash()
```

Value

(chr) en-dash character

See Also

Other marks: `mid_dot()`, `thin_space()`

Examples

```
en_dash()
```

`fmt_magnitude` *Format magnitude*

Description

Format a numeric vector into a string with specified magnitude (billion, million, thousand).

Usage

```
fmt_magnitude(
  x,
  metric,
  rate_unit = NULL,
  digits = 1,
  nsmall = 1,
  decimal.mark = ".",
  big.mark = ",",
  mag = NULL,
  count_label_thousands = FALSE,
  assert_prop_range = TRUE
)
```

Arguments

`x`	(num) numeric vector
`metric`	(chr ‘c("prop", "pp", "count", "rate")‘)
`rate_unit`	(chr: default NULL) unit label for rates (e.g., "deaths", "cases"). Required when metric = "rate", ignored otherwise.
`digits`	(int: default 1L) passed to ‘round()‘
`nsmall`	(int: default 1L) passed to ‘format()‘
`decimal.mark`	(chr: default ".") decimal mark passed to ‘format()‘
`big.mark`	(chr: default ",") thousands separator passed to ‘format()‘
`mag`	(chr: default NULL) magnitude override - see set_magnitude() - For props/pp: "as-is" (no scaling, use values as provided) - For counts: "t" (thousand), "m" (million), "b" (billion) - For rates: "per10", "per100", "per1k", ..., "per10b"
`count_label_thousands`	(lgl: default FALSE) allow thousands magnitude? Not Lancet-valid. Passed to ‘set_magnitude()‘
`assert_prop_range`	(lgl: default TRUE) assert that proportion values are between -1 and +1 (or -100 and +100 if mag="as-is")

Details

Unaware of styles, just a hard-coded git-er-done function.

Caution - thousands magnitude is not Lancet compliant.

Value

(chr) formatted string

See Also

Other vector_formats: `add_epsilon()`, `format_oxford_comma()`, `fround()`, `fround_count_rate()`, `fround_metric()`, `fround_metric_lancet()`, `fround_props()`

Other magnitudes: `set_magnitude()`, `set_magnitude_count()`, `set_magnitude_prop()`, `set_magnitude_rate()`

Examples

```
fmt_magnitude(123456789, metric = "count")
fmt_magnitude(0.0000123, metric = "rate", rate_unit = "deaths")
```

---

format_journal_clu *Format central, lower, upper value triplets for journal presentation*

Description

Defaults are generic. This function allows special formtting marks to be applied by journal. Use ‘format_lancet_clu()‘ for Lancet-specific formatting. Use ‘format_nature_clu()‘ for Nature-specific formatting.

Usage

```
format_journal_clu(
  central,
  lower,
  upper,
  metric,
  rate_unit = NULL,
  mag = NULL,
  style_name = "nature"
)
```

Arguments

central	(num) central, point_estimate value vector
lower	(num) lower bound vector
upper	(num) upper bound vector
metric	(chr c(prop, pp, count, rate)) metric - proportion, percentage point, count, or rate

rate_unit	(chr: default NULL) rate unit label (required when metric = 'rate')
mag	(chr: default NULL) magnitude override - see set_magnitude() - For props/pp: "as-is" (no scaling, use values as provided) - For counts: "t" (thousand), "m" (million), "b" (billion) - For rates: "per10", "per100", "per1k", ..., "per10b" - Examples: "deaths", "cases", "events", "births"
style_name	(chr: default 'nature') style name - controls rounding and formatting.

Details

Takes three vectors as main arguments for data.table-friendly vectorization.

'central' could be mean, median, point_estimate

Transform c(central = 0.994, lower = 0.984, upper = 0.998) to "99.4% (98.4–99.8)"

Accounts for negative values, and UIs that cross zero. Checks if central, lower, upper values are in the correct order.

Value

(chr) formatted string vector

See Also

Other styled_formats: format_journal_df(), format_lancet_clu(), format_lancet_df(), format_metric_cols(), format_nature_clu(), format_nature_df(), fround_clu_triplet(), new_style()

Examples

```
format_journal_clu(
 central = c(0.994, -0.994)
 , lower = c(0.984, -0.998)
 , upper = c(0.998, -0.984)
 , metric = "prop"
)

# Rate formatting with rate_unit
format_journal_clu(
  central   = 0.0000123,
  lower     = 0.0000098,
  upper     = 0.0000152,
  metric    = "rate",
  rate_unit = "deaths"
)
```

`format_journal_df` *Return a table with formatted central, lower, upper*

Description

Assumes a single data-type (metric) for the whole table (e.g. 'prop', 'pp', 'count')

Usage

```
format_journal_df(
  df,
  metric,
  new_var = "clu_fmt",
  style_name = "nature",
  central_var = "mean",
  lower_var = "lower",
  upper_var = "upper",
  remove_clu_columns = TRUE,
  rate_unit = NULL,
  mag = NULL
)
```

Arguments

`df`	(data.frame, data.table)
`metric`	(chr c('prop', 'pp', 'count', 'rate')) a single metric
`new_var`	(chr: default 'clu_fmt') name of new formatted column
`style_name`	(chr: default 'nature') style name - controls rounding and formatting.
`central_var`	(chr: default 'mean') name of central tendency variable
`lower_var`	(chr: default 'lower') name of lower bound variable
`upper_var`	(chr: default 'upper') name of upper bound variable
`remove_clu_columns`	(lgl: default TRUE) remove central, lower, upper variables after formatting?
`rate_unit`	(chr: default NULL) rate unit label (required when metric = 'rate') - Examples: "deaths", "cases", "events", "births"
`mag`	(chr: default NULL) magnitude override - see set_magnitude() - For props/pp: "as-is" (no scaling, use values as provided) - For counts: "t" (thousand), "m" (million), "b" (billion) - For rates: "per10", "per100", "per1k", ..., "per10b" - Examples: "deaths", "cases", "events", "births"

Value

(data.frame) data.frame, data.table with new 'clu_fmt' column

See Also

Other styled_formats: format_journal_clu(), format_lancet_clu(), format_lancet_df(), format_metric_cols(), format_nature_clu(), format_nature_df(), fround_clu_triplet(), new_style()

Examples

```
df <- data.frame(
 location_id = c(1, 2, 3)
 , mean      = c(0.1234, 0, -0.3456)
 , lower     = c(0.1134, -0.2245, -0.4445)
 , upper     = c(0.1334, 0.2445, 0.3556)
)
format_journal_df(df, metric = "prop")

# Rate formatting example
rate_df <- data.frame(
  location = c("Global", "USA"),
  mean     = c(0.0000123, 0.0000456),
  lower    = c(0.0000098, 0.0000401),
  upper    = c(0.0000152, 0.0000512)
)
format_journal_df(rate_df, metric = "rate", rate_unit = "deaths")
```

---

format_lancet_clu *Format central, lower, upper value triplets for Lancet journal presentation*

Description

Format central, lower, upper value triplets for Lancet journal presentation

Usage

```
format_lancet_clu(central, lower, upper, metric, rate_unit = NULL, mag = NULL)
```

Arguments

central	(num) central, point_estimate value vector
lower	(num) lower bound vector
upper	(num) upper bound vector
metric	(chr c(prop, pp, count, rate)) metric - proportion, percentage point, count, or rate
rate_unit	(chr: default NULL) rate unit label (required when metric = 'rate')
mag	(chr: default NULL) magnitude override - see set_magnitude() - For props/pp: "as-is" (no scaling, use values as provided) - For counts: "t" (thousand), "m" (million), "b" (billion) - For rates: "per10", "per100", "per1k", ..., "per10b" - Examples: "deaths", "cases", "events", "births"

Value

(chr) formatted string vector

See Also

Other styled_formats: `format_journal_clu()`, `format_journal_df()`, `format_lancet_df()`, `format_metric_cols()`, `format_nature_clu()`, `format_nature_df()`, `fround_clu_triplet()`, `new_style()`

Examples

```
format_lancet_clu(
   central  = c(0.994, -0.994)
   , lower  = c(0.984, -0.998)
   , upper  = c(0.998, -0.984)
   , metric = "prop"
)

# Rate example with Lancet formatting
format_lancet_clu(
  central   = 0.0000123,
  lower     = 0.0000098,
  upper     = 0.0000152,
  metric    = "rate",
  rate_unit = "deaths"
)
```

---

format_lancet_df *Return a table with formatted central, lower, upper for Lancet journal*

Description

Assumes a single data-type (metric) for the whole table (e.g. 'prop', 'pp', 'count')

Usage

```
format_lancet_df(
  df,
  metric,
  new_var = "clu_fmt",
  central_var = "mean",
  lower_var = "lower",
  upper_var = "upper",
  remove_clu_columns = TRUE,
  rate_unit = NULL,
  mag = NULL
)
```

Arguments

df	(data.table) with central, lower, upper columns
metric	(chr c('prop', 'pp', 'count', 'rate')) metric - proportion, percentage point, count, or rate
new_var	(chr: default 'clu_fmt') name of new formatted column
central_var	(chr: default 'mean') name of central tendency e.g. 'point_estimate'
lower_var	(chr: default 'lower')
upper_var	(chr: default 'upper')
remove_clu_columns	(lgl: default TRUE) remove central, lower, upper columns after formatting?
rate_unit	(chr: default NULL) rate unit label (required when metric = 'rate')
mag	(chr: default NULL) magnitude override - see set_magnitude() - For props/pp: "as-is" (no scaling, use values as provided) - For counts: "t" (thousand), "m" (million), "b" (billion) - For rates: "per10", "per100", "per1k", ..., "per10b" - Examples: "deaths", "cases", "events", "births"

Value

(data.frame, data.table) with mean_95_UI_formatted column, and central, lower, upper columns removed (if specified)

See Also

Other styled_formats: `format_journal_clu()`, `format_journal_df()`, `format_lancet_clu()`, `format_metric_cols()`, `format_nature_clu()`, `format_nature_df()`, `fround_clu_triplet()`, `new_style()`

Examples

```
df <- data.frame(
   location_did    = 1
   , location_name = "Global"
   , me_name       = "vacc_dpt1"
   , mean          = 55.8e6
   , lower         = 50.7e6
   , upper         = 60.7e6
)
format_lancet_df(df = df, metric = "count", central_var = 'mean')
```

`format_metric_cols` *Format multiple data.frame 'mean_*' columns for presentation (by metric).*

Description

Format one or more 'mean_' columns by magnitude, metric, and style.

Usage

```
format_metric_cols(
  df,
  metric,
  var_prefix = "mean",
  rate_unit = NULL,
  mag = NULL,
  style_name = "nature"
)
```

Arguments

`df`	(data.table) input data.table with one or more 'mean_' columns
`metric`	(chr c('prop', 'pp', 'count', 'rate')) a single metric
`var_prefix`	(chr: default 'mean') prefix of mean variable names to format. Implemented as e.g. "^mean[_]+" to capture 'mean', 'mean_1990', 'mean_2000', etc.
`rate_unit`	(chr: default NULL) unit label for rates (e.g., "deaths", "cases"). Required when metric = "rate", ignored otherwise.
`mag`	(chr: default NULL) magnitude override - see set_magnitude() - For props/pp: "as-is" (no scaling, use values as provided) - For counts: "t" (thousand), "m" (million), "b" (billion) - For rates: "per10", "per100", "per1k", ..., "per10b"
`style_name`	(chr: default 'nature') style name - controls rounding and formatting.

Value

(data.table) copy of input data.table with formatted mean column(s)

See Also

Other styled_formats: `format_journal_clu()`, `format_journal_df()`, `format_lancet_clu()`, `format_lancet_df()`, `format_nature_clu()`, `format_nature_df()`, `fround_clu_triplet()`, `new_style()`

Examples

```
df <- data.frame(
  location_id = c(1, 2, 3)
  , mean_1990 = c(100, 1e6, 1e9)
  , mean_2000 = c(200, 2e6, 2e-1)
 )
format_metric_cols(df, metric = "count")
```

`format_nature_clu` *Format central, lower, upper value triplets for Nature journal presentation*

Description

Format central, lower, upper value triplets for Nature journal presentation

Usage

```
format_nature_clu(central, lower, upper, metric, rate_unit = NULL, mag = NULL)
```

Arguments

`central`	(num) central, point_estimate value vector
`lower`	(num) lower bound vector
`upper`	(num) upper bound vector
`metric`	(chr c(prop, pp, count, rate)) metric - proportion, percentage point, count, or rate
`rate_unit`	(chr: default NULL) rate unit label (required when metric = 'rate')
`mag`	(chr: default NULL) magnitude override - see set_magnitude() - For props/pp: "as-is" (no scaling, use values as provided) - For counts: "t" (thousand), "m" (million), "b" (billion) - For rates: "per10", "per100", "per1k", ..., "per10b" - Examples: "deaths", "cases", "events", "births"

Value

(chr) formatted string vector

See Also

Other styled_formats: `format_journal_clu()`, `format_journal_df()`, `format_lancet_clu()`, `format_lancet_df()`, `format_metric_cols()`, `format_nature_df()`, `fround_clu_triplet()`, `new_style()`

Examples

```
format_nature_clu(
   central  = c(0.994, -0.994)
   , lower  = c(0.984, -0.998)
   , upper  = c(0.998, -0.984)
   , metric = "prop"
)

# Rate example with Nature formatting
format_nature_clu(
  central   = 0.0000123,
  lower     = 0.0000098,
  upper     = 0.0000152,
  metric    = "rate",
  rate_unit = "cases"
)
```

`format_nature_df` *Return a table with formatted central, lower, upper for Nature journal*

Description

Return a table with formatted central, lower, upper for Nature journal

Usage

```
format_nature_df(
  df,
  metric,
  new_var = "clu_fmt",
  central_var = "mean",
  lower_var = "lower",
  upper_var = "upper",
  remove_clu_columns = TRUE,
  rate_unit = NULL,
  mag = NULL
)
```

Arguments

`df`	(data.table)
`metric`	(chr c('prop', 'pp', 'count', 'rate')) a single metric
`new_var`	(chr: default 'clu_fmt') name of new formatted column
`central_var`	(chr: default 'mean') name of central tendency variable
`lower_var`	(chr: default 'lower') name of lower bound variable
`upper_var`	(chr: default 'upper') name of upper bound variable

remove_clu_columns
: (lgl: default TRUE) remove central, lower, upper columns after formatting?

rate_unit
: (chr: default NULL) rate unit label (required when metric = 'rate')

mag
: (chr: default NULL) magnitude override - see set_magnitude() - For props/pp: "as-is" (no scaling, use values as provided) - For counts: "t" (thousand), "m" (million), "b" (billion) - For rates: "per10", "per100", "per1k", ..., "per10b" - Examples: "deaths", "cases", "events", "births"

Value

(data.table) copy of input data.table with new 'clu_fmt' column

See Also

Other styled_formats: `format_journal_clu()`, `format_journal_df()`, `format_lancet_clu()`, `format_lancet_df()`, `format_metric_cols()`, `format_nature_clu()`, `fround_clu_triplet()`, `new_style()`

Examples

```
df <- data.frame(
   location_did    = 1
   , location_name = "Global"
   , me_name       = "vacc_dpt1"
   , mean          = 55.8e6
   , lower         = 50.7e6
   , upper         = 60.7e6
)
format_nature_df(df = df, metric = "count", central_var = 'mean')
```

---

format_oxford_comma *Format vector of items with Oxford comma*

Description

Format vector of items with Oxford comma

Usage

```
format_oxford_comma(vec, sep = "and")
```

Arguments

vec
: (any) vector of items to format

sep
: (chr: default "and") separator before last item

Value

(chr) formatted string with Oxford comma

See Also

Other vector_formats: add_epsilon(), fmt_magnitude(), fround(), fround_count_rate(), fround_metric(), fround_metric_lancet(), fround_props()

Examples

```
format_oxford_comma(1:2)
format_oxford_comma(1:3)
format_oxford_comma(1:3, "or")
```

---

fround *Format and round*

Description

Unaware of data-type or schema, just a hard-coded git-er-done function.

Usage

```
fround(x, digits = 1L, nsmall = 1L, decimal.mark = ".")
```

Arguments

x	(num) numeric vector
digits	(integer) passed to 'round()'
nsmall	(integer) passed to 'format()'
decimal.mark	(chr) passed to 'format()'

Value

(chr) formatted string

See Also

Other vector_formats: add_epsilon(), fmt_magnitude(), format_oxford_comma(), fround_count_rate(), fround_metric(), fround_metric_lancet(), fround_props()

Examples

```
fround(0.123456789)
fround(0.123456789, digits = 3)
fround(0.123456789, digits = 3, nsmall = 4)
```

fround_metric *Format and round with data-type suffix*

Description

Unaware of schema, just a hard-coded git-er-done function.

Usage

```
fround_metric(x, metric = "prop", digits = 1L, nsmall = 1L, decimal.mark = ".")
```

Arguments

x	(num) numeric value
metric	(chr: default 'prop' c('prop', 'pp', 'count', 'rate'))
digits	(integer: default 1L) passed to 'round()'
nsmall	(integer: default 1L) passed to 'format()'
decimal.mark	(chr: default ".") decimal mark passed to 'format()'

Value

(chr) formatted string

See Also

Other vector_formats: add_epsilon(), fmt_magnitude(), format_oxford_comma(), fround(), fround_count_rate(), fround_metric_lancet(), fround_props()

Examples

```
fround_metric(0.123456789)
fround_metric(0.123456789, 'pp', 3, 4)
fround_metric(c(55.8346, 123.456789), 'count', 3, 4, ".")
```

fround_metric_lancet *Format and round with data-type suffix*

Description

Lancet-specific wrapper for 'fround_metric()', using mid-dot as decimal mark. Retaining for legacy purposes (no Nature equivalent)

Usage

```
fround_metric_lancet(
  x,
  metric = "prop",
  digits = 1L,
  nsmall = 1L,
  decimal.mark = mid_dot()
)
```

Arguments

x	(num) numeric value
metric	(chr: default 'prop' c('prop', 'pp', 'count', rate))
digits	(integer: default 1L) passed to 'round()'
nsmall	(integer: default 1L) passed to 'format()'
decimal.mark	(chr: default mid_dot()) decimal mark passed to 'format()'

Value

(chr) formatted string

See Also

Other vector_formats: add_epsilon(), fmt_magnitude(), format_oxford_comma(), fround(), fround_count_rate(), fround_metric(), fround_props()

Examples

```
fround_metric_lancet(0.123456789)
fround_metric_lancet(0.123456789, 'pp', 3, 4)
fround_metric_lancet(c(55.8346, 123.456789), 'count', 3, 4, ".")
```

---

get_metric_labels *Get metric labels*

Description

Centrally managed definition for all metric labels.

Usage

```
get_metric_labels(metric)
```

Arguments

metric	(chr)

Value

(list) named list of metric labels

See Also

Other metrics: `get_metrics()`

Examples

```
get_metric_labels('prop')
```

`get_metrics` *Get metrics*

Description

Centrally managed definition for all allowed metrics.

Usage

```
get_metrics()
```

Value

(chr) vector of allowed metrics

See Also

Other metrics: `get_metric_labels()`

Examples

```
get_metrics()
```

get_style *Get a style from the styles dictionary*

Description

Accessor function to retrieve a style from the package's styles dictionary.

Usage

```
get_style(style_name)
```

Arguments

style_name	(chr) name of the style to retrieve

Value

(list) the requested style as a named list

See Also

Other styles: assert_style_schema(), get_style_schema(), new_style(), set_style(), style_lancet(), style_nature()

Examples

```
get_style("lancet")
```

get_style_schema *Get style schema*

Description

Centrally managed definition for all required journal format styles.

Usage

```
get_style_schema()
```

Value

(list) named list of style elements and their expected types

See Also

Other styles: assert_style_schema(), get_style(), new_style(), set_style(), style_lancet(), style_nature()

Examples

```
get_style_schema()
```

`mid_dot` *Mid-dot*

Description

Lancet numeric decimal standard: "·" ("\U00B7")

Usage

```
mid_dot()
```

Value

(chr) mid-dot character

See Also

Other marks: `en_dash()`, `thin_space()`

Examples

```
mid_dot()
```

`new_style` *Make a new style by args*

Description

Wrapper function to create and set a new style in one step.

Usage

```
new_style(
  style_name,
  prop_digits_round = 1,
  prop_nsmall = 1,
  prop_invert_all_neg_UI = TRUE,
  count_method = "sigfig",
  count_digits_sigfig = 3,
  count_pad_sigfigs = TRUE,
  count_nsmall = 1,
  count_big.mark = ",",
  count_label_thousands = FALSE,
```

```
  count_invert_all_neg_UI = FALSE,
  rate_method = "sigfig",
  rate_digits_sigfig = 3,
  rate_pad_sigfigs = TRUE,
  rate_nsmall = 1,
  rate_invert_all_neg_UI = FALSE,
  decimal.mark = ".",
  neg_mark_mean = "-",
  neg_mark_UI = "-",
  UI_only = FALSE,
  UI_text = "",
  is_lancet = FALSE,
  round_5_up = TRUE,
  assert_clu_order = TRUE,
  assert_prop_range = TRUE
)
```

Arguments

style_name	(chr) name of the style to set
prop_digits_round	(int: default 1) number of digits to round proportions to
prop_nsmall	(int: default 1) minimum number of digits to the right of the decimal point - proportions
prop_invert_all_neg_UI	(lgl: default TRUE) if mean/lower/upper are all negative, invert signs in UI for proportions? e.g. -0.1 (-0.2 to -0.05) becomes - 0.1 (0.05 to 0.2)
count_method	(chr: c("sigfig", "decimal", "int")) choose how to report counts - prioritize sigfigs across mean/lower/upper, hard-set decimals, or leave numbers in integer space.
count_digits_sigfig	(int: default 3) number of significant figures for counts
count_pad_sigfigs	(lgl: default TRUE) signif(5.00, 3) is "5" - do you want to pad the trailing 0s back on - usually TRUE?
count_nsmall	(int: default 1) passed to ‘format()‘ if ‘count_method‘ == ’decimal’
count_big.mark	(chr: default ",") character to use for counts thousand, million, billion separator e.g. ","
count_label_thousands	(lgl: default FALSE) whether format counts as e.g. 10,000 as ’10 thousand’
count_invert_all_neg_UI	(lgl: default FALSE) if mean/lower/upper are all negative, invert signs in UI for counts? e.g. -10 (-20 to -5) becomes -10 (5 to 20)
rate_method	(chr: c("sigfig", "decimal", "int")) choose how to report rates - prioritize sigfigs across mean/lower/upper, hard-set decimals, or leave numbers in integer space.
rate_digits_sigfig	(int: default 3) number of significant figures for rates

`rate_pad_sigfigs`
: (lgl: default TRUE) signif(5.00, 3) is "5" - do you want to pad the trailing 0s back on for rates - usually TRUE?

`rate_nsmall`
: (int: default 1) passed to ‘format()‘ if ‘rate_method‘ == ’decimal’

`rate_invert_all_neg_UI`
: (lgl: default FALSE) if mean/lower/upper are all negative, invert signs in UI for rates? e.g. -10 (-20 to -5) per million becomes -10 (5 to 20) per million

`decimal.mark`
: (chr: default ".") decimal mark e.g. "." or ‘mid_dot()‘ for Lancet.

`neg_mark_mean`
: (chr: default "-") string to describe central value negatives - e.g. "-1 (-2 to 4)" could become "Negtive 1 (-2 to 4)"

`neg_mark_UI`
: (chr: default "-") string to describe negative sign in UI brackets e.g. "1 (-2 to 4)" could become "1 (–2 to 4)" (en-dash)

`UI_only`
: (lgl: default FALSE) Return only UI from ‘format_journal_df()‘ family functions?

`UI_text`
: (chr: default "") Text to appear inside UI brackets before numbers e.g. "2 (1 – 4)" could become "2 (95%UI 1 – 4)"

`is_lancet`
: (lgl: default FALSE) TRUE to handle edge-case Lancet count formatting policies

`round_5_up`
: (lgl: default TRUE) In R, ‘round(1245, 3)‘ is "1240". Do you want to round to "1250" instead? Default TRUE to conform with common expectations.

`assert_clu_order`
: (lgl: default TRUE) whether to assert CLU relationships (ensure lower < central < upper)

`assert_prop_range`
: (lgl: default TRUE) whether to assert that proportion values are between -1 and +1

Value

(chr) invisible vector of input objects

See Also

Other styles: `assert_style_schema()`, `get_style()`, `get_style_schema()`, `set_style()`, `style_lancet()`, `style_nature()`

Other styled_formats: `format_journal_clu()`, `format_journal_df()`, `format_lancet_clu()`, `format_lancet_df()`, `format_metric_cols()`, `format_nature_clu()`, `format_nature_df()`, `fround_clu_triplet()`

Examples

```
new_style(style_name = "my_style")
```

set_magnitude *Define magnitude, magnitude label and denominator for a vector of numeric values*

Description

Support function used on _central_ (e.g. mean) values for later formatting. Routes to appropriate helper based on metric.

Usage

```
set_magnitude(
  x,
  metric,
  mag = NULL,
  count_label_thousands = FALSE,
  assert_prop_range = TRUE,
  verbose = TRUE
)
```

Arguments

x	(num) numeric vector
metric	(chr) metric: "prop", "pp", "count", "rate" (required)
mag	(chr: default NULL) magnitude override (NULL = auto-detect) - For counts: "t", "m", "b" - For rates: "per10", "per100", "per1k", ..., "per10b" - For props/pp: "as-is" (no scaling, use values as provided)
count_label_thousands	(lgl: default FALSE) allow "thousands" magnitude for counts? Not Lancet-valid.
assert_prop_range	(lgl: default TRUE) for proportions, assert that values are between -1 and +1 (or -100 and +100 if mag="as-is")?
verbose	(lgl: default TRUE) show warnings?

Value

(data.frame) with columns: mag, mag_label, denom

See Also

Other magnitudes: fmt_magnitude(), set_magnitude_count(), set_magnitude_prop(), set_magnitude_rate()

Examples

```
# Proportions (no scaling)
set_magnitude(c(0.5, 0.75), metric = "prop")

# Counts
set_magnitude(c(1e3, 1e6, 1e9), metric = "count")

# Rates
set_magnitude(c(0.0000123, 0.0000456), metric = "rate")
```

`set_style` *Set a new style by list*

Description

Set a new style by list

Usage

```
set_style(style_name, style_entry)
```

Arguments

`style_name`	(chr) name of the style to set
`style_entry`	(list) named list representing the style entry

Value

(chr) invisible vector of input objects, to allow easier un-locking

See Also

Other styles: `assert_style_schema()`, `get_style()`, `get_style_schema()`, `new_style()`, `style_lancet()`, `style_nature()`

Examples

```
set_style(
   style_name    = "my_style"
   , style_entry = list(
      style_name               = "my_style"
      , prop_digits_round      = 1
      , prop_nsmall            = 1
      , prop_invert_all_neg_UI = TRUE
      , count_method           = "sigfig"
      , count_digits_sigfig    = 3
      , count_pad_sigfigs      = TRUE
      , count_nsmall           = 1
      , count_big.mark         = ","
```

```
        , count_label_thousands   = FALSE
        , count_invert_all_neg_UI = FALSE
        , rate_method             = "sigfig"
        , rate_digits_sigfig      = 3
        , rate_pad_sigfigs        = TRUE
        , rate_nsmall             = 1
        , rate_invert_all_neg_UI  = FALSE
        , decimal.mark            = "."
        , neg_mark_mean           = "-"
        , neg_mark_UI             = "-"
        , UI_only                 = FALSE
        , UI_text                 = ""
        , is_lancet               = FALSE
        , round_5_up              = TRUE
        , assert_clu_order        = TRUE
        , assert_prop_range       = TRUE
    )
)
```

`style_lancet` *Lancet style schema*

Description

Pre-defined style schema for Lancet journal formatting

Usage

```
style_lancet()
```

Value

(list) named list representing the lancet style

See Also

Other styles: `assert_style_schema()`, `get_style()`, `get_style_schema()`, `new_style()`, `set_style()`, `style_nature()`

Examples

```
style_lancet()
```

style_nature *Nature style schema*

Description

The default style for the package.

Usage

```
style_nature()
```

Details

Pre-defined style schema for Nature journal formatting.

Value

(list) named list representing the nature style

See Also

Other styles: `assert_style_schema()`, `get_style()`, `get_style_schema()`, `new_style()`, `set_style()`, `style_lancet()`

Examples

```
style_nature()
```

thin_space *Thin space*

Description

Lancet thin space separator for counts 10,000 – 999,9999 instead of comma ",": ("\U2009")

Usage

```
thin_space()
```

Value

(chr) thin space character

See Also

Other marks: `en_dash()`, `mid_dot()`

Examples

```
thin_space()
```

Index

∗ **magnitudes**
  fmt_magnitude, 3
  set_magnitude, 22
∗ **marks**
  en_dash, 2
  mid_dot, 19
  thin_space, 25
∗ **metrics**
  get_metric_labels, 16
  get_metrics, 17
∗ **styled_formats**
  format_journal_clu, 4
  format_journal_df, 6
  format_lancet_clu, 7
  format_lancet_df, 8
  format_metric_cols, 10
  format_nature_clu, 11
  format_nature_df, 12
  new_style, 19
∗ **styles**
  get_style, 18
  get_style_schema, 18
  new_style, 19
  set_style, 23
  style_lancet, 24
  style_nature, 25
∗ **vector_formats**
  fmt_magnitude, 3
  format_oxford_comma, 13
  fround, 14
  fround_metric, 15
  fround_metric_lancet, 15

add_epsilon, *4*, *14–16*
assert_style_schema, *18*, *21*, *23–25*

en_dash, 2, *19*, *25*

fmt_magnitude, 3, *14–16*, *22*
format_journal_clu, 4, *7–11*, *13*, *21*
format_journal_df, *5*, 6, *8–11*, *13*, *21*
format_lancet_clu, *5*, 7, *7*, *9–11*, *13*, *21*
format_lancet_df, *5*, *7*, 8, *8*, *10, 11*, *13*, *21*
format_metric_cols, *5*, *7–9*, 10, *11*, *13*, *21*
format_nature_clu, *5*, *7–10*, 11, *13*, *21*
format_nature_df, *5*, *7–11*, 12, *21*
format_oxford_comma, *4*, 13, *14–16*
fround, *4*, 14, *14–16*
fround_clu_triplet, *5*, *7–11*, *13*, *21*
fround_count_rate, *4*, *14–16*
fround_metric, *4*, *14*, 15, *16*
fround_metric_lancet, *4*, *14*, 15, *15*
fround_props, *4*, *14–16*

get_metric_labels, 16, *17*
get_metrics, 17, *17*
get_style, 18, *18*, *21*, *23–25*
get_style_schema, 18, *18*, *21*, *23–25*

mid_dot, *2*, 19, *25*

new_style, *5*, *7–11*, *13*, *18*, 19, *23–25*

set_magnitude, *4*, 22
set_magnitude_count, *4*, *22*
set_magnitude_prop, *4*, *22*
set_magnitude_rate, *4*, *22*
set_style, *18*, *21*, 23, *24, 25*
style_lancet, *18*, *21*, *23*, 24, *25*
style_nature, *18*, *21*, *23, 24*, 25

thin_space, *2*, *19*, 25