Package: csstyle (via r-universe)

July 2, 2026

Title Standardizing Outputs in the 'Core Surveillance' Style

Version 2026.7.1

Description A comprehensive styling system for consistent visual outputs using 'Core Surveillance' guidelines. Provides 'ggplot2' themes, color palettes, and dual formatting functions to standardize graphs, tables, and reports. Includes Norwegian formatting conventions for domestic use and international journal standards for academic publications. Features number formatting (percentages, per-100k rates, log transformations), date formatting (Norwegian vs ISO 8601), and a hierarchical color system. Focuses on consistency over flexibility to ensure reproducible, professional outputs across research contexts.

Depends R (>= 3.3.0)

License MIT + file LICENSE

URL https://niphr.github.io/csstyle/, https://github.com/niphr/csstyle

BugReports https://github.com/niphr/csstyle/issues

Encoding UTF-8

Imports data.table, ggplot2, glue, magrittr, ragg, rstudioapi, stringr, csdata

Suggests testthat, knitr, rmarkdown, lubridate

VignetteBuilder knitr

Config/Needs/website niphr/cstemplate

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libfontconfig1-dev libfreetype6-dev libfribidi-dev libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libwebp-dev

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

Date/Publication 2026-07-01 17:48:43 UTC

RemoteUrl https://github.com/niphr/csstyle

RemoteRef HEAD

RemoteSha ab63a3619765d43721cee9ced6e9c01934aabd9c

Contents

`as_github_code` *Create a github_code object from a GitHub URL*

Description

Reads the lines of a file on GitHub and returns them as a `github_code` object. A "blob" URL is rewritten to the corresponding raw URL before the file is read. The object can be printed with line numbers and an optional link back to GitHub (see `print.github_code`).

Usage

```
as_github_code(url)
```

Arguments

`url`	Character string. A GitHub URL to a file, either a "blob" URL (e.g. `https://github.com/owner/repo/b` or the corresponding raw URL.

Details

Create a github_code object from a GitHub URL

Value

An object of class `github_code`: a character vector of the file's lines with a `pretty_url` attribute.

See Also

`print.github_code`

Examples

```
## Not run:
# Requires an internet connection
x <- as_github_code("https://github.com/niphr/csstyle/blob/main/README.md")
print(x, lines = 1:3)

## End(Not run)
```

`colors` *Color definitions for Core Surveillance styling*

Description

A list holding the named colors, the base color, and the predefined color palettes used by the Core Surveillance styling functions.

Usage

```
colors
```

Format

A list with the following elements:

named_colors Named character vector of hex codes "H1" to "H12".

base The base color (element "H1" of `named_colors`).

palettes Named list of palettes. The `primary` palettes have 1 to 12 levels (`primary_1` to `primary_12`); `warning_3` has 3 levels; `posneg_1` and `posneg_2` have 1 and 2 levels.

palette_names Character vector of the palette family names ("primary", "posneg", "warning").

Value

A list. See the Format section for the elements.

See Also

Other csstyle_utilities: `display_all_palettes()`, `every_nth()`, `set_global()`

Examples

```
# Access named colors
colors$named_colors["H1"]

# View primary palette
colors$palettes$primary_3

# See all available palettes
names(colors$palettes)
```

---

display_all_palettes *Display all available color palettes*

Description

Creates a visualization showing all available Core Surveillance color palettes with their color codes and names.

Usage

```
display_all_palettes()
```

Value

A ggplot2 plot displaying all color palettes

See Also

Other csstyle_utilities: `colors`, `every_nth()`, `set_global()`

Examples

```
# Display all available color palettes
display_all_palettes()
```

every_nth *Returns every nth discrete value*

Description

Creates a function that returns every nth element from a vector, useful for axis breaks in ggplot2.

Usage

```
every_nth(n)
```

Arguments

n	Integer specifying which nth value to return (e.g., n=2 returns every 2nd value)

Value

A function that returns every nth discrete value.

See Also

Other csstyle_utilities: colors, display_all_palettes(), set_global()

Examples

```
every_nth(4)(c(1:10))
## Not run:
scale_x_discrete(NULL, breaks = every_nth(n = 2))

## End(Not run)
```

format_cstidy_age_as_factor *Format cstidy age codes as factor with Norwegian text*

Description

Converts age codes (e.g., "009", "005_014") to readable Norwegian age labels with proper formatting.

Converts age codes to readable Norwegian age labels and returns as character vector.

Usage

```
format_cstidy_age_as_factor(x)

format_cstidy_age_as_character(x)
```

Arguments

x	Character vector containing age codes or age group codes

Value

Factor vector with Norwegian age labels (e.g., "9 år", "5-14 år")

Character vector with Norwegian age labels (e.g., "9 år", "5-14 år")

Examples

```
csstyle::format_cstidy_age_as_factor("009")
csstyle::format_cstidy_age_as_factor("005-014")
csstyle::format_cstidy_age_as_character("009")
csstyle::format_cstidy_age_as_character("005-014")
```

---

format_date_as_journal

Format date and datetime using journal conventions (ISO 8601)

Description

Functions for formatting dates and datetimes using ISO 8601 journal conventions.

Usage

```
format_date_as_journal(x = lubridate::today(), format = "%Y-%m-%d")

format_datetime_as_journal(
  x = lubridate::now(),
  format = "%Y-%m-%d %H:%M:%S"
)

format_datetime_as_journal_file(
  x = lubridate::now(),
  format = "%Y_%m_%d_%H%M%S"
)
```

Arguments

x	Date or datetime object to format
format	Character string specifying the desired format

Value

Character vector with journal-formatted dates/datetimes

Examples

```
# Format current date/time using journal conventions (ISO 8601)
format_date_as_journal()
format_datetime_as_journal()

# Format specific dates
format_date_as_journal(as.Date("2023-12-25"))
format_datetime_as_journal_file(as.POSIXct("2023-12-25 14:30:00"))
```

`format_date_as_nor` *Format date and datetime using Norwegian conventions*

Description

Functions for formatting dates and datetimes using Norwegian conventions or for safe filename use.

Usage

```
format_date_as_nor(x = lubridate::today(), format = "%d.%m.%Y")

format_datetime_as_nor(
  x = lubridate::now(),
  format = "%d.%m.%Y kl. %H:%S"
)

format_datetime_as_file(x = lubridate::now(), format = "%Y_%m_%d_%H%M%S")
```

Arguments

`x`	Date or datetime object to format
`format`	Character string specifying the desired format

Value

Character vector with formatted dates/datetimes

Examples

```
# Format current date/time using Norwegian conventions
format_date_as_nor()
format_datetime_as_nor()

# Format specific dates
format_date_as_nor(as.Date("2023-12-25"))
format_datetime_as_file(as.POSIXct("2023-12-25 14:30:00"))
```

format_location_code_as_factor *Convert Norwegian location codes to factor with readable labels*

Description

Converts Norwegian location codes (municipality/county codes) to factors with human-readable Norwegian location names.

Converts Norwegian location codes to character vector with human-readable Norwegian location names.

Usage

```
format_location_code_as_factor(
  x,
  label = NULL,
  label_if_not_unique = NULL,
  reference = csdata::nor_locations_names(),
  direction = 1
)

format_location_code_as_character(
  x,
  label = NULL,
  label_if_not_unique = NULL,
  reference = csdata::nor_locations_names(),
  direction = 1
)
```

Arguments

x	Character vector of Norwegian location codes
label	Column name from reference data to use for labels (default: uses global setting)
label_if_not_unique	Column name to use when location names are not unique (default: uses global setting)
reference	Data.table with location reference data (default: csdata::nor_locations_names())
direction	Integer: 1 for normal order, -1 for reversed order (default: 1)

Value

Factor vector with Norwegian location names as labels

Character vector with Norwegian location names

Examples

```
# Convert location codes to factors (requires csdata package)
location_codes <- c("03", "11", "15")
format_location_code_as_factor(location_codes)
format_location_code_as_character(location_codes)
```

`format_num_as_journal_invlog10_1` *Format numbers on inverse log-10 scale with journal conventions (1 digit)*

Description

Formats values by applying 10^x transformation then formatting with journal conventions and 1 decimal place.

Formats values by applying 10^x transformation then formatting with journal conventions and 2 decimal places.

Usage

```
format_num_as_journal_invlog10_1(x)

format_num_as_journal_invlog10_2(x)
```

Arguments

x	Numeric vector (log-10 scale values)

Value

Character vector with journal-formatted numbers

Character vector with journal-formatted numbers

See Also

Other csstyle_journal_formatters: `format_num_as_journal_invlog2_1()`, `format_num_as_journal_invloge_1()`, `format_num_as_journal_num_0()`, `format_num_as_journal_per100k_0()`, `format_num_as_journal_perc_0()`

Examples

```
# Transform log10 values back to original scale
format_num_as_journal_invlog10_1(c(1, 2, 3))  # 10^1, 10^2, 10^3
format_num_as_journal_invlog10_2(c(1, 2, 3))
```

format_num_as_journal_invlog2_1 *Format numbers on inverse log-2 scale with journal conventions (1 digit)*

Description

Formats values by applying 2^x transformation then formatting with journal conventions and 1 decimal place.

Formats values by applying 2^x transformation then formatting with journal conventions and 2 decimal places.

Usage

```
format_num_as_journal_invlog2_1(x)

format_num_as_journal_invlog2_2(x)
```

Arguments

x	Numeric vector (log-2 scale values)

Value

Character vector with journal-formatted numbers

Character vector with journal-formatted numbers

See Also

Other csstyle_journal_formatters: format_num_as_journal_invlog10_1(), format_num_as_journal_invloge_1(), format_num_as_journal_num_0(), format_num_as_journal_per100k_0(), format_num_as_journal_perc_0()

Examples

```
# Transform log2 values back to original scale
format_num_as_journal_invlog2_1(c(1, 2, 3))  # 2^1, 2^2, 2^3
format_num_as_journal_invlog2_2(c(1, 2, 3))
```

format_num_as_journal_invloge_1 *Format numbers on inverse natural log scale with journal conventions (1 digit)*

Description

Formats values by applying exp(x) transformation then formatting with journal conventions and 1 decimal place.

Formats values by applying exp(x) transformation then formatting with journal conventions and 2 decimal places.

Usage

```
format_num_as_journal_invloge_1(x)

format_num_as_journal_invloge_2(x)
```

Arguments

x	Numeric vector (natural log scale values)

Value

Character vector with journal-formatted numbers

Character vector with journal-formatted numbers

See Also

Other csstyle_journal_formatters: format_num_as_journal_invlog10_1(), format_num_as_journal_invlog2_1(), format_num_as_journal_num_0(), format_num_as_journal_per100k_0(), format_num_as_journal_perc_0()

Examples

```
# Transform natural log values back to original scale
format_num_as_journal_invloge_1(c(0, 1, 2))  # exp(0), exp(1), exp(2)
format_num_as_journal_invloge_2(c(0, 1, 2))
```

`format_num_as_journal_num_0` *Format numbers with journal conventions (0 digits)*

Description

Formats numeric values using journal number conventions with 0 decimal places.

Formats numeric values using journal number conventions with 1 decimal place.

Formats numeric values using journal number conventions with 2 decimal places.

Usage

```
format_num_as_journal_num_0(x)

format_num_as_journal_num_1(x)

format_num_as_journal_num_2(x)
```

Arguments

x	Numeric vector to format

Value

Character vector with journal-formatted numbers

Character vector with journal-formatted numbers

Character vector with journal-formatted numbers

See Also

Other csstyle_journal_formatters: `format_num_as_journal_invlog10_1()`, `format_num_as_journal_invlog2_1()`, `format_num_as_journal_invloge_1()`, `format_num_as_journal_per100k_0()`, `format_num_as_journal_perc_0()`

Examples

```
# Journal number formatting
format_num_as_journal_num_0(c(1234.56, 9876.54, NA))
format_num_as_journal_num_1(c(1234.56, 9876.54, NA))
format_num_as_journal_num_2(c(1234.56, 9876.54, NA))
```

`format_num_as_journal_per100k_0` *Format numbers per 100k with journal conventions (0 digits)*

Description

Formats numeric values as "per 100k" rates using journal number conventions with 0 decimal places.

Formats numeric values as "per 100k" rates using journal number conventions with 1 decimal place.

Formats numeric values as "per 100k" rates using journal number conventions with 2 decimal places.

Usage

```
format_num_as_journal_per100k_0(x)

format_num_as_journal_per100k_1(x)

format_num_as_journal_per100k_2(x)
```

Arguments

x	Numeric vector to format

Value

Character vector with journal-formatted numbers and "/100k" suffix

Character vector with journal-formatted numbers and "/100k" suffix

Character vector with journal-formatted numbers and "/100k" suffix

See Also

Other csstyle_journal_formatters: `format_num_as_journal_invlog10_1()`, `format_num_as_journal_invlog2_1()`, `format_num_as_journal_invloge_1()`, `format_num_as_journal_num_0()`, `format_num_as_journal_perc_0()`

Examples

```
# Format rates per 100k population
format_num_as_journal_per100k_0(c(123.45, 678.90, NA))
format_num_as_journal_per100k_1(c(123.45, 678.90, NA))
format_num_as_journal_per100k_2(c(123.45, 678.90, NA))
```

`format_num_as_journal_perc_0` *Format numbers as percentages with journal conventions (0 digits)*

Description

Formats numeric values as percentages using journal number conventions with 0 decimal places.

Formats numeric values as percentages using journal number conventions with 1 decimal place.

Formats numeric values as percentages using journal number conventions with 2 decimal places.

Usage

```
format_num_as_journal_perc_0(x)

format_num_as_journal_perc_1(x)

format_num_as_journal_perc_2(x)
```

Arguments

x	Numeric vector to format

Value

Character vector with journal-formatted numbers and "

Character vector with journal-formatted numbers and "

Character vector with journal-formatted numbers and "

See Also

Other csstyle_journal_formatters: `format_num_as_journal_invlog10_1()`, `format_num_as_journal_invlog2_1()`, `format_num_as_journal_invloge_1()`, `format_num_as_journal_num_0()`, `format_num_as_journal_per100k_0()`

Examples

```
# Format percentages
format_num_as_journal_perc_0(c(12.34, 56.78, NA))
format_num_as_journal_perc_1(c(12.34, 56.78, NA))
format_num_as_journal_perc_2(c(12.34, 56.78, NA))
```

format_num_as_nor_invlog10_1 *Format numbers on inverse log-10 scale with Norwegian conventions (1 digit)*

Description

Formats values by applying 10^x transformation then formatting with Norwegian conventions and 1 decimal place.

Formats values by applying 10^x transformation then formatting with Norwegian conventions and 2 decimal places.

Usage

```
format_num_as_nor_invlog10_1(x)

format_num_as_nor_invlog10_2(x)
```

Arguments

x	Numeric vector (log-10 scale values)

Value

Character vector with Norwegian-formatted numbers

Character vector with Norwegian-formatted numbers

See Also

Other csstyle_formatters: format_num_as_nor_invlog2_1(), format_num_as_nor_invloge_1(), format_num_as_nor_num_0(), format_num_as_nor_per100k_0(), format_num_as_nor_perc_0()

Examples

```
# Transform log10 values back to original scale
format_num_as_nor_invlog10_1(c(1, 2, 3))  # 10^1, 10^2, 10^3
format_num_as_nor_invlog10_2(c(1, 2, 3))
```

format_num_as_nor_invlog2_1 *Format numbers on inverse log-2 scale with Norwegian conventions (1 digit)*

Description

Formats values by applying 2^x transformation then formatting with Norwegian conventions and 1 decimal place.

Formats values by applying 2^x transformation then formatting with Norwegian conventions and 2 decimal places.

Usage

```
format_num_as_nor_invlog2_1(x)

format_num_as_nor_invlog2_2(x)
```

Arguments

x	Numeric vector (log-2 scale values)

Value

Character vector with Norwegian-formatted numbers

Character vector with Norwegian-formatted numbers

See Also

Other csstyle_formatters: format_num_as_nor_invlog10_1(), format_num_as_nor_invloge_1(), format_num_as_nor_num_0(), format_num_as_nor_per100k_0(), format_num_as_nor_perc_0()

Examples

```
# Transform log2 values back to original scale
format_num_as_nor_invlog2_1(c(1, 2, 3))  # 2^1, 2^2, 2^3
format_num_as_nor_invlog2_2(c(1, 2, 3))
```

format_num_as_nor_invloge_1 *Format numbers on inverse natural log scale with Norwegian conventions (1 digit)*

Description

Formats values by applying exp(x) transformation then formatting with Norwegian conventions and 1 decimal place.

Formats values by applying exp(x) transformation then formatting with Norwegian conventions and 2 decimal places.

Usage

```
format_num_as_nor_invloge_1(x)

format_num_as_nor_invloge_2(x)
```

Arguments

x	Numeric vector (natural log scale values)

Value

Character vector with Norwegian-formatted numbers

Character vector with Norwegian-formatted numbers

See Also

Other csstyle_formatters: format_num_as_nor_invlog10_1(), format_num_as_nor_invlog2_1(), format_num_as_nor_num_0(), format_num_as_nor_per100k_0(), format_num_as_nor_perc_0()

Examples

```
# Transform natural log values back to original scale
format_num_as_nor_invloge_1(c(0, 1, 2))  # exp(0), exp(1), exp(2)
format_num_as_nor_invloge_2(c(0, 1, 2))
```

`format_num_as_nor_num_0` *Format numbers with Norwegian conventions (0 digits)*

Description

Formats numeric values using Norwegian number conventions with 0 decimal places.

Formats numeric values using Norwegian number conventions with 1 decimal place.

Formats numeric values using Norwegian number conventions with 2 decimal places.

Usage

```
format_num_as_nor_num_0(x)

format_num_as_nor_num_1(x)

format_num_as_nor_num_2(x)
```

Arguments

x	Numeric vector to format

Value

Character vector with Norwegian-formatted numbers

Character vector with Norwegian-formatted numbers

Character vector with Norwegian-formatted numbers

See Also

Other csstyle_formatters: `format_num_as_nor_invlog10_1()`, `format_num_as_nor_invlog2_1()`, `format_num_as_nor_invloge_1()`, `format_num_as_nor_per100k_0()`, `format_num_as_nor_perc_0()`

Examples

```
# Basic number formatting
format_num_as_nor_num_0(c(1234.56, 9876.54, NA))
format_num_as_nor_num_1(c(1234.56, 9876.54, NA))
format_num_as_nor_num_2(c(1234.56, 9876.54, NA))
```

`format_num_as_nor_per100k_0` *Format numbers per 100k with Norwegian conventions (0 digits)*

Description

Formats numeric values as "per 100k" rates using Norwegian number conventions with 0 decimal places.

Formats numeric values as "per 100k" rates using Norwegian number conventions with 1 decimal place.

Formats numeric values as "per 100k" rates using Norwegian number conventions with 2 decimal places.

Usage

```
format_num_as_nor_per100k_0(x)

format_num_as_nor_per100k_1(x)

format_num_as_nor_per100k_2(x)
```

Arguments

x	Numeric vector to format

Value

Character vector with Norwegian-formatted numbers and "/100k" suffix

Character vector with Norwegian-formatted numbers and "/100k" suffix

Character vector with Norwegian-formatted numbers and "/100k" suffix

See Also

Other csstyle_formatters: `format_num_as_nor_invlog10_1()`, `format_num_as_nor_invlog2_1()`, `format_num_as_nor_invloge_1()`, `format_num_as_nor_num_0()`, `format_num_as_nor_perc_0()`

Examples

```
# Format rates per 100k population
format_num_as_nor_per100k_0(c(123.45, 678.90, NA))
format_num_as_nor_per100k_1(c(123.45, 678.90, NA))
format_num_as_nor_per100k_2(c(123.45, 678.90, NA))
```

`format_num_as_nor_perc_0` *Format numbers as percentages with Norwegian conventions (0 digits)*

Description

Formats numeric values as percentages using Norwegian number conventions with 0 decimal places.

Formats numeric values as percentages using Norwegian number conventions with 1 decimal place.

Formats numeric values as percentages using Norwegian number conventions with 2 decimal places.

Usage

```
format_num_as_nor_perc_0(x)

format_num_as_nor_perc_1(x)

format_num_as_nor_perc_2(x)
```

Arguments

x	Numeric vector to format

Value

Character vector with Norwegian-formatted numbers and "

Character vector with Norwegian-formatted numbers and "

Character vector with Norwegian-formatted numbers and "

See Also

Other csstyle_formatters: `format_num_as_nor_invlog10_1()`, `format_num_as_nor_invlog2_1()`, `format_num_as_nor_invloge_1()`, `format_num_as_nor_num_0()`, `format_num_as_nor_per100k_0()`

Examples

```
# Format percentages
format_num_as_nor_perc_0(c(12.34, 56.78, NA))
format_num_as_nor_perc_1(c(12.34, 56.78, NA))
format_num_as_nor_perc_2(c(12.34, 56.78, NA))
```

pretty_breaks *Pretty breaks*

Description

Creates a function for generating pretty axis breaks with Norwegian number formatting.

Usage

```
pretty_breaks(n = 5, formatting_fn = csstyle::format_num_as_nor_num_0, ...)
```

Arguments

n	Number of desired breaks (default: 5)
formatting_fn	Formatting function to apply to break labels (default: Norwegian number formatting)
...	Additional arguments passed to pretty()

Value

A function that can be used for breaks on graphing.

Examples

```
library(ggplot2)

# Use pretty breaks with default formatting
ggplot(mtcars, aes(x = mpg, y = hp)) +
  geom_point() +
  scale_x_continuous(breaks = pretty_breaks(n = 4))
```

print.github_code *Print a github_code object*

Description

Prints the lines of a `github_code` object with line numbers. Use the `lines` argument to print a subset of lines, and `include_url` or `include_url_as_link` to also print a link back to the file on GitHub.

Usage

```
## S3 method for class 'github_code'
print(x, ...)
```

Arguments

x	A github_code object, as created by as_github_code.
...	Further arguments controlling the output: lines (integer vector of line numbers to print), include_url (logical; print the GitHub link), and include_url_as_link (logical; print the link as HTML).

Details

Print a github_code object

Value

The input x, invisibly.

See Also

as_github_code

Examples

```
## Not run:
# Requires an internet connection
x <- as_github_code("https://github.com/niphr/csstyle/blob/main/README.md")
print(x, lines = 1:3, include_url = TRUE)

## End(Not run)
```

---

save_a4 *Save ggplot in A4 scale*

Description

Saves a ggplot2 plot with A4 paper dimensions.

Usage

```
save_a4(q, filename, landscape = T, scaling_factor = 1)
```

Arguments

q	ggplot2 plot object to save
filename	Character string specifying the output filename (with extension)
landscape	Logical indicating if plot should use landscape orientation (default: TRUE)
scaling_factor	Numeric scaling factor for A4 dimensions (default: 1)

Details

PNG output is rendered with the ‘ragg‘ AGG device (‘ragg::agg_png‘), which is markedly faster than the default ‘grDevices‘ PNG device for busy, many-geom ggplots (and gives cleaner text). Non-PNG filenames fall back to ‘ggsave‘’s extension-based device selection.

Value

Nothing (called for side effects).

Examples

```
library(ggplot2)

# Create a plot
p <- ggplot(mtcars, aes(x = mpg, y = hp)) +
  geom_point() +
  theme_cs()

## Not run:
# Save in landscape A4
save_a4(p, "myplot.png")

# Save in portrait A4 with larger scaling
save_a4(p, "myplot_large.png", landscape = FALSE, scaling_factor = 1.5)

## End(Not run)
```

---

scale_color_cs *ggplot2 color scale using Core Surveillance color palettes*

Description

Creates a discrete color scale for ggplot2 using predefined Core Surveillance color palettes.

Usage

```
scale_color_cs(..., palette = "primary", direction = 1)
```

Arguments

...	Additional arguments passed to ggplot2::discrete_scale()
palette	Name of color palette to use (default: "primary"). Options: "primary", "warning", "posneg"
direction	Direction of color palette: 1 for normal, -1 for reversed (default: 1)

Value

A ggplot2 discrete color scale

See Also

Other csstyle_ggplot2: scale_fill_cs(), theme_cs()

Examples

```
library(ggplot2)

# Default primary color scale
ggplot(mtcars, aes(x = mpg, y = hp, color = factor(cyl))) +
  geom_point() +
  scale_color_cs()

# Use warning palette
ggplot(mtcars, aes(x = mpg, y = hp, color = factor(cyl))) +
  geom_point() +
  scale_color_cs(palette = "warning")
```

---

scale_fill_cs *ggplot2 fill scale using Core Surveillance color palettes*

Description

Creates a discrete fill scale for ggplot2 using predefined Core Surveillance color palettes.

Usage

```
scale_fill_cs(..., palette = "primary", direction = 1)
```

Arguments

...	Additional arguments passed to ggplot2::discrete_scale()
palette	Name of color palette to use (default: "primary"). Options: "primary", "warning", "posneg"
direction	Direction of color palette: 1 for normal, -1 for reversed (default: 1)

Value

A ggplot2 discrete fill scale

See Also

Other csstyle_ggplot2: scale_color_cs(), theme_cs()

Examples

```
library(ggplot2)

# Default primary fill scale
ggplot(mtcars, aes(x = factor(cyl), fill = factor(cyl))) +
  geom_bar() +
  scale_fill_cs()

# Use posneg palette
ggplot(mtcars, aes(x = factor(vs), fill = factor(vs))) +
  geom_bar() +
  scale_fill_cs(palette = "posneg")
```

`set_global` *Set global configuration for location code formatting*

Description

Sets the global defaults that control how location codes are turned into labels by `format_location_code_as_factor` and `format_location_code_as_character`. These defaults are used whenever those functions are called without an explicit `label` or `label_if_not_unique`.

Usage

```
set_global(
  location_code_to_factor_label = "location_name",
  location_code_to_factor_label_if_not_unique = "location_name_description_nb"
)
```

Arguments

`location_code_to_factor_label`
: Column name in the reference data to use for location labels (default: "location_name").

`location_code_to_factor_label_if_not_unique`
: Column name in the reference data to use when the chosen labels are not unique (default: "location_name_description_nb").

Value

Called for its side effect of updating the global configuration. Returns the assigned value invisibly.

See Also

Other csstyle_utilities: `colors`, `display_all_palettes()`, `every_nth()`

Examples

```
# Set global location formatting preferences
set_global(
  location_code_to_factor_label = "custom_name",
  location_code_to_factor_label_if_not_unique = "custom_description"
)

# Reset to defaults
set_global()
```

---

theme_cs *Core Surveillance ggplot2 theme for consistent styling*

Description

Custom ggplot2 theme following Core Surveillance visual guidelines with configurable grid lines and formatting.

Returns a ggplot2 theme component that rotates the x-axis tick labels to be vertical. Add it to a plot like any other theme element.

Usage

```
theme_cs(
  base_size = 16,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22,
  legend_position = "right",
  x_axis_vertical = FALSE,
  panel_on_top = TRUE,
  panel.grid.major.x = FALSE,
  panel.grid.minor.x = FALSE,
  panel.grid.major.y = TRUE,
  panel.grid.minor.y = TRUE
)

set_x_axis_vertical()
```

Arguments

base_size	Base font size in points (default: 16)
base_family	Base font family (default: "")
base_line_size	Base line size relative to base_size (default: base_size/22)
base_rect_size	Base rectangle size relative to base_size (default: base_size/22)
legend_position	Position of legend: "right", "left", "top", "bottom", or "none" (default: "right")

`x_axis_vertical`
: Logical: should x-axis labels be vertical? (default: FALSE)

`panel_on_top`
: Logical: should panel be drawn on top of data? (default: TRUE)

`panel.grid.major.x`
: Logical or element: major vertical grid lines (default: FALSE)

`panel.grid.minor.x`
: Logical or element: minor vertical grid lines (default: FALSE)

`panel.grid.major.y`
: Logical or element: major horizontal grid lines (default: TRUE)

`panel.grid.minor.y`
: Logical or element: minor horizontal grid lines (default: TRUE)

Value

Complete ggplot2 theme object

A ggplot2 theme component that can be added to a plot.

See Also

Other csstyle_ggplot2: `scale_color_cs()`, `scale_fill_cs()`

Examples

```
library(ggplot2)

# Basic usage with default settings
ggplot(mtcars, aes(x = mpg, y = hp)) +
  geom_point() +
  theme_cs()

# With custom settings
ggplot(mtcars, aes(x = mpg, y = hp)) +
  geom_point() +
  theme_cs(legend_position = "bottom", x_axis_vertical = TRUE)
library(ggplot2)

# Add vertical x-axis labels to any plot
ggplot(mtcars, aes(x = rownames(mtcars), y = mpg)) +
  geom_col() +
  set_x_axis_vertical()
```

`view_html` *View HTML in the RStudio viewer*

Description

Opens HTML content in the RStudio viewer pane.

Usage

```
view_html(x)
```

Arguments

x	Character string containing HTML content to display

Value

Nothing (called for side effects).

Examples

```
# View simple HTML content
html_content <- "<h1>Hello World</h1><p>This is a test.</p>"
## Not run:
view_html(html_content)

## End(Not run)
```

Index