Package: gtscales (via r-universe)

May 16, 2026

Title Color Scale Extensions for 'gt'

Version 0.0.1

Depends R (>= 4.1.0)

Description Adds legend helpers for color-encoded 'gt' tables. The primary interface combines 'gt::data_color()' with matched legends for continuous, binned, quantile, and discrete scales, while lower-level helpers support legend-only and reusable scale specification workflows.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports gt, rlang, scales

Suggests knitr, rmarkdown, testthat (>= 3.0.0), vdiffr

Config/testthat/edition 3

URL https://christophertkenny.com/gtscales/

VignetteBuilder knitr

Config/Needs/website christopherkenny/ctktemplate

Config/pak/sysreqs cmake make libuv1-dev libxml2-dev libssl-dev libnode-dev

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

Date/Publication 2026-04-16 18:07:24 UTC

RemoteUrl https://github.com/christopherkenny/gtscales

RemoteRef HEAD

RemoteSha ceb60ec95b4a35b0ce61d82e02d6632547c8c42d

Contents

`gtscale_apply` *Apply only the color component of a* `gtscales` *spec*

Description

Apply only the color component of a `gtscales` spec

Usage

```
gtscale_apply(data, spec)
```

Arguments

`data`	A `gt_tbl` created by `gt::gt()`.
`spec`	A `gtscale_spec`.

Value

A modified gt::gt table.

gtscale_apply_legend *Apply a* gtscales *spec and attach its legend*

Description

Apply a gtscales spec and attach its legend

Usage

```
gtscale_apply_legend(data, spec)
```

Arguments

data	A gt_tbl created by gt::gt().
spec	A gtscale_spec.

Value

A modified gt::gt table.

Examples

```
library(gt)

spec <- gtscale_spec_continuous(
  num,
  palette = c('#A0442C', 'white', '#0063B1'),
  title = 'Value'
)

exibble |>
  gt() |>
  gtscale_apply_legend(spec)
```

gtscale_color_bins *Add only a binned color legend to a* gt *table*

Description

This is a lower-level helper for cases where table coloring is already handled elsewhere. For the usual "color and legendize" workflow, prefer gtscale_data_color_bins().

Usage

```
gtscale_color_bins(
  data,
  column = NULL,
  palette,
  domain = NULL,
  bins = NULL,
  transform = NULL,
  oob = NULL,
  right = FALSE,
  labels = NULL,
  title = NULL,
  width = "180px",
  height = "14px"
)
```

Arguments

data	A gt_tbl created by gt::gt().
column	An optional numeric, Date, POSIXt, or difftime column in the underlying table used to infer domain.
palette	A vector of colors, palette endpoints, a single named palette, or a palette function used to color the bins.
domain	A vector of length 2 giving the scale limits. When omitted, the limits are inferred from column.
bins	Optional bin boundaries or a break function. When omitted, breaks are generated from domain, column, and transform.
transform	A transformation specification understood by scales::as.transform(). This is used when generating default bins or when interpreting break functions.
oob	Out-of-bounds handling function or shortcut. Use a function like scales::oob_squish() or a shortcut such as "censor" or "squish".
right	Whether intervals should be closed on the right. The default of FALSE yields intervals like [a, b).
labels	An optional labeling function or a character vector for the bins. When a function is supplied, it is applied to the bin boundaries before interval labels are constructed.
title	Optional legend title.
width	Width of the legend.
height	Height of the swatches.

Value

A modified gt::gt table.

Examples

```
library(gt)

exibble |>
  gt() |>
  data_color(
    columns = currency,
    method = 'bin',
    palette = c('#f7fbff', '#08306b'),
    bins = c(0, 10, 100, 1000, 10000000)
  ) |>
  gtscale_color_bins(
    column = currency,
    palette = c('#f7fbff', '#08306b'),
    bins = c(0, 10, 100, 1000, 10000000),
    title = 'Binned values'
  )
```

---

gtscale_color_continuous

Add only a continuous color legend to a `gt` *table*

Description

This is a lower-level helper for cases where table coloring is already handled elsewhere. For the usual "color and legendize" workflow, prefer `gtscale_data_color_continuous()`.

Usage

```
gtscale_color_continuous(
  data,
  column = NULL,
  palette = NULL,
  domain = NULL,
  breaks = NULL,
  labels = NULL,
  title = NULL,
  transform = NULL,
  oob = NULL,
  direction = "to right",
  width = "160px",
  height = "14px",
  fn = NULL
)
```

Arguments

data	A gt_tbl created by gt::gt().
column	An optional numeric column in the underlying table used to infer domain.
palette	A vector of colors used in the legend gradient. A single named palette or palette function can also be supplied.
domain	A numeric vector of length 2 giving the scale limits. When omitted, the limits are inferred from column.
breaks	Optional break values or a break function to display below the gradient.
labels	An optional labeling function or a character vector for the breaks.
title	Optional legend title.
transform	A transformation specification understood by scales::as.transform(). When omitted, an appropriate transform is inferred from the data.
oob	Out-of-bounds handling function or shortcut. Use a function like scales::oob_squish() or a shortcut such as "censor", "squish", "keep", or "discard".
direction	CSS gradient direction. Defaults to "to right".
width	Width of the legend bar.
height	Height of the legend bar.
fn	Backward-compatible fallback for passing a scales palette function. palette is preferred.

Value

A modified gt::gt table.

Examples

```
library(gt)

exibble |>
  gt() |>
  data_color(
    columns = num,
    method = 'numeric',
    palette = c('#A0442C', 'white', '#0063B1')
  ) |>
  gtscale_color_continuous(
    column = num,
    palette = c('#A0442C', 'white', '#0063B1'),
    title = 'Value'
  )
```

gtscale_color_discrete *Add only a discrete color legend to a* gt *table*

Description

This is a lower-level helper for cases where table coloring is already handled elsewhere. For the usual "color and legendize" workflow, prefer gtscale_data_color_discrete().

Usage

```
gtscale_color_discrete(
  data,
  values,
  labels = values,
  title = NULL,
  swatch_size = "12px"
)
```

Arguments

data	A gt_tbl created by gt::gt().
values	A vector of color values or a single named discrete palette.
labels	Labels for each color swatch. Defaults to values.
title	Optional legend title.
swatch_size	Size of each discrete color swatch.

Value

A modified gt::gt table.

Examples

```
library(gt)

data.frame(
  category = c('Low', 'Medium', 'High'),
  value = c(12, 47, 83)
) |>
  gt() |>
  data_color(
    columns = category,
    method = 'factor',
    palette = c('#1b9e77', '#d95f02', '#7570b3')
  ) |>
  gtscale_color_discrete(
    values = c('#1b9e77', '#d95f02', '#7570b3'),
```

```
  labels = c('Low', 'Medium', 'High'),
  title = 'Category'
)
```

---

gtscale_color_diverging *Add only a diverging color legend to a* gt *table*

Description

This is a lower-level helper for cases where table coloring is already handled elsewhere. For the usual "color and legendize" workflow, prefer `gtscale_data_color_diverging()`.

Usage

```
gtscale_color_diverging(
  data,
  column = NULL,
  palette,
  domain = NULL,
  midpoint = 0,
  breaks = NULL,
  labels = NULL,
  title = NULL,
  transform = NULL,
  oob = NULL,
  direction = "to right",
  width = "160px",
  height = "14px",
  mid_color = "#FFFFFF"
)
```

Arguments

data	A gt_tbl created by `gt::gt()`.
column	An optional numeric column or shared set of numeric columns in the underlying table used to infer domain.
palette	Two endpoint colors, three diverging colors, or a single named palette.
domain	A numeric vector of length 2 giving the scale limits. When omitted, the limits are inferred from column.
midpoint	Numeric midpoint used to anchor the diverging scale.
breaks	Optional break values or a break function to display below the gradient.
labels	An optional labeling function or a character vector for the breaks.
title	Optional legend title.

transform	A transformation specification understood by scales::as.transform(). When omitted, an appropriate transform is inferred from the data.
oob	Out-of-bounds handling function or shortcut. Use a function like scales::oob_squish() or a shortcut such as "censor", "squish", "keep", or "discard".
direction	CSS gradient direction. Defaults to "to right".
width	Width of the legend bar.
height	Height of the legend bar.
mid_color	Midpoint color when palette supplies only two endpoint colors.

Value

A modified gt::gt table.

`gtscale_color_quantiles`

Add only a quantile color legend to a `gt` *table*

Description

This is a lower-level helper for cases where table coloring is already handled elsewhere. For the usual "color and legendize" workflow, prefer gtscale_data_color_quantiles().

Usage

```
gtscale_color_quantiles(
  data,
  column,
  palette,
  quantiles = 4,
  oob = NULL,
  right = FALSE,
  labels = NULL,
  title = NULL,
  width = "180px",
  height = "14px"
)
```

Arguments

data	A gt_tbl created by gt::gt().
column	A numeric, Date, POSIXt, or difftime column in the underlying table used to infer quantile boundaries.
palette	A vector of colors, palette endpoints, a single named palette, or a palette function used to color the quantile bins.
quantiles	The number of quantile groups.

oob	Out-of-bounds handling function or shortcut. Use a function like scales::oob_squish() or a shortcut such as "censor" or "squish".
right	Whether intervals should be closed on the right. The default of FALSE yields intervals like [a, b).
labels	An optional labeling function or a character vector for the quantile ranges. When a function is supplied, it is applied to the quantile boundaries before interval labels are constructed.
title	Optional legend title.
width	Width of the legend.
height	Height of the swatches.

Value

A modified gt::gt table.

Examples

```
library(gt)

exibble |>
  gt() |>
  data_color(
    columns = num,
    method = 'quantile',
    palette = c('#fdd49e', '#fdbb84', '#ef6548', '#990000'),
    quantiles = 4
  ) |>
  gtscale_color_quantiles(
    column = num,
    palette = c('#fdd49e', '#fdbb84', '#ef6548', '#990000'),
    quantiles = 4,
    title = 'Quantile bins'
  )
```

---

gtscale_data_color_bins

Color a numeric gt *column and add a matching binned legend*

Description

This is the primary interface for binned scales in gtscales.

Usage

```
gtscale_data_color_bins(
  data,
  column,
  palette,
  bins = NULL,
  domain = NULL,
  transform = NULL,
  oob = NULL,
  right = FALSE,
  labels = NULL,
  title = NULL,
  width = "180px",
  height = "14px",
  apply_to = c("fill", "text"),
  na_color = NULL,
  alpha = NULL,
  reverse = FALSE,
  autocolor_text = TRUE,
  contrast_algo = c("apca", "wcag"),
  autocolor_light = "#FFFFFF",
  autocolor_dark = "#000000"
)
```

Arguments

data	A gt_tbl created by gt::gt().
column	A numeric, Date, POSIXt, or difftime column to color and legendize.
palette	A vector of colors, palette endpoints, a single named palette, or a palette function used for the bins.
bins	Optional bin boundaries or a break function. When omitted, breaks are generated from domain, column, and transform.
domain	A vector of length 2 giving the scale limits. When omitted, the limits are inferred from column.
transform	A transformation specification understood by scales::as.transform(). This is used when generating default bins or when interpreting break functions.
oob	Out-of-bounds handling function or shortcut. Use a function like scales::oob_squish() or a shortcut such as "censor" or "squish".
right	Whether intervals should be closed on the right. The default of FALSE yields intervals like [a, b).
labels	An optional labeling function or a character vector for the bins. When a function is supplied, it is applied to the bin boundaries before interval labels are constructed.
title	Optional legend title.
width	Width of the legend.

height	Height of the swatches.
apply_to	Whether colors should be applied to cell fill or text.
na_color	Color used for missing values.
alpha	Alpha applied by gt::data_color().
reverse	Whether to reverse the color mapping.
autocolor_text	Whether to automatically adjust text color.
contrast_algo	Contrast algorithm passed to gt::data_color().
autocolor_light	Light text color used by gt::data_color().
autocolor_dark	Dark text color used by gt::data_color().

Value

A modified gt::gt table.

Examples

```
library(gt)

exibble |>
  gt() |>
  gtscale_data_color_bins(
    column = currency,
    palette = c('#f7fbff', '#08306b'),
    bins = c(0, 10, 100, 1000, 10000, 70000),
    title = 'Currency bins'
  )
```

---

gtscale_data_color_continuous

Color a numeric gt *column and add a matching continuous legend*

Description

This is the primary interface for continuous scales in gtscales.

Usage

```
gtscale_data_color_continuous(
  data,
  column,
  palette = NULL,
  domain = NULL,
  breaks = NULL,
  labels = NULL,
  title = NULL,
```

```
  transform = NULL,
  oob = NULL,
  direction = "to right",
  width = "160px",
  height = "14px",
  apply_to = c("fill", "text"),
  na_color = NULL,
  alpha = NULL,
  reverse = FALSE,
  autocolor_text = TRUE,
  contrast_algo = c("apca", "wcag"),
  autocolor_light = "#FFFFFF",
  autocolor_dark = "#000000",
  fn = NULL
)
```

Arguments

data	A gt_tbl created by gt::gt().
column	A numeric column to color and legendize.
palette	A vector of colors used in the table and legend gradient. A single named palette or palette function can also be supplied.
domain	A numeric vector of length 2 giving the scale limits. When omitted, the limits are inferred from column.
breaks	Optional break values or a break function to display below the gradient.
labels	An optional labeling function or a character vector for the breaks.
title	Optional legend title.
transform	A transformation specification understood by scales::as.transform(). When omitted, an appropriate transform is inferred from the data.
oob	Out-of-bounds handling function or shortcut. Use a function like scales::oob_squish() or a shortcut such as "censor", "squish", "keep", or "discard".
direction	CSS gradient direction. Defaults to "to right".
width	Width of the legend bar.
height	Height of the legend bar.
apply_to	Whether colors should be applied to cell fill or text.
na_color	Color used for missing values.
alpha	Alpha applied by gt::data_color().
reverse	Whether to reverse the color mapping.
autocolor_text	Whether to automatically adjust text color.
contrast_algo	Contrast algorithm passed to gt::data_color().
autocolor_light	Light text color used by gt::data_color().
autocolor_dark	Dark text color used by gt::data_color().
fn	Optional scales function passed to gt::data_color(). For the legend, palette is preferred because it is more reliable than inspecting closure internals.

Value

A modified gt::gt table.

Examples

```
library(gt)

exibble |>
  gt() |>
  gtscale_data_color_continuous(
    column = num,
    palette = c('#A0442C', 'white', '#0063B1'),
    title = 'Value'
  )
```

`gtscale_data_color_discrete`

Color a categorical `gt` *column and add a matching discrete legend*

Description

This is the primary interface for discrete scales in `gtscales`.

Usage

```
gtscale_data_color_discrete(
  data,
  column,
  values,
  labels = values,
  title = NULL,
  swatch_size = "12px",
  levels = NULL,
  ordered = FALSE,
  na_color = NULL,
  alpha = NULL,
  reverse = FALSE,
  apply_to = c("fill", "text"),
  autocolor_text = TRUE,
  contrast_algo = c("apca", "wcag"),
  autocolor_light = "#FFFFFF",
  autocolor_dark = "#000000"
)
```

Arguments

data	A gt_tbl created by gt::gt().
column	A categorical column to color and legendize.
values	A vector of color values or a single named discrete palette used in the table and legend.
labels	Labels for each color swatch. Defaults to values.
title	Optional legend title.
swatch_size	Size of each discrete color swatch.
levels	Optional factor levels passed to gt::data_color().
ordered	Whether the scale should be treated as ordered.
na_color	Color used for missing values.
alpha	Alpha applied by gt::data_color().
reverse	Whether to reverse the color mapping.
apply_to	Whether colors should be applied to cell fill or text.
autocolor_text	Whether to automatically adjust text color.
contrast_algo	Contrast algorithm passed to gt::data_color().
autocolor_light	Light text color used by gt::data_color().
autocolor_dark	Dark text color used by gt::data_color().

Value

A modified gt::gt table.

Examples

```
library(gt)

data.frame(
  category = c('Low', 'Medium', 'High'),
  value = c(12, 47, 83)
) |>
  gt() |>
  gtscale_data_color_discrete(
    column = category,
    values = c('#1b9e77', '#d95f02', '#7570b3'),
    labels = c('Low', 'Medium', 'High'),
    title = 'Category'
  )
```

`gtscale_data_color_diverging` *Color a numeric* `gt` *column with a diverging scale and add a matching legend*

Description

This is the primary interface for midpoint-aware diverging scales in `gtscales`.

Usage

```
gtscale_data_color_diverging(
  data,
  column,
  palette,
  domain = NULL,
  midpoint = 0,
  breaks = NULL,
  labels = NULL,
  title = NULL,
  transform = NULL,
  oob = NULL,
  direction = "to right",
  width = "160px",
  height = "14px",
  apply_to = c("fill", "text"),
  na_color = NULL,
  alpha = NULL,
  reverse = FALSE,
  autocolor_text = TRUE,
  contrast_algo = c("apca", "wcag"),
  autocolor_light = "#FFFFFF",
  autocolor_dark = "#000000",
  mid_color = "#FFFFFF"
)
```

Arguments

`data`	A `gt_tbl` created by `gt::gt()`.
`column`	A numeric column or shared set of numeric columns to color and legendize.
`palette`	Two endpoint colors, three diverging colors, or a single named palette.
`domain`	A numeric vector of length 2 giving the scale limits. When omitted, the limits are inferred from `column`.
`midpoint`	Numeric midpoint used to anchor the diverging scale.
`breaks`	Optional break values or a break function to display below the gradient.

labels	An optional labeling function or a character vector for the breaks.
title	Optional legend title.
transform	A transformation specification understood by scales::as.transform(). When omitted, an appropriate transform is inferred from the data.
oob	Out-of-bounds handling function or shortcut. Use a function like scales::oob_squish() or a shortcut such as "censor", "squish", "keep", or "discard".
direction	CSS gradient direction. Defaults to "to right".
width	Width of the legend bar.
height	Height of the legend bar.
apply_to	Whether colors should be applied to cell fill or text.
na_color	Color used for missing values.
alpha	Alpha applied by gt::data_color().
reverse	Whether to reverse the color mapping.
autocolor_text	Whether to automatically adjust text color.
contrast_algo	Contrast algorithm passed to gt::data_color().
autocolor_light	Light text color used by gt::data_color().
autocolor_dark	Dark text color used by gt::data_color().
mid_color	Midpoint color when palette supplies only two endpoint colors.

Value

A modified gt::gt table.

`gtscale_data_color_quantiles` *Color a numeric* `gt` *column and add a matching quantile legend*

Description

This is the primary interface for quantile scales in `gtscales`.

Usage

```
gtscale_data_color_quantiles(
  data,
  column,
  palette,
  quantiles = 4,
  oob = NULL,
  right = FALSE,
  labels = NULL,
  title = NULL,
```

```
  width = "180px",
  height = "14px",
  apply_to = c("fill", "text"),
  na_color = NULL,
  alpha = NULL,
  reverse = FALSE,
  autocolor_text = TRUE,
  contrast_algo = c("apca", "wcag"),
  autocolor_light = "#FFFFFF",
  autocolor_dark = "#000000"
)
```

Arguments

data	A gt_tbl created by gt::gt().
column	A numeric, Date, POSIXt, or difftime column to color and legendize.
palette	A vector of colors, palette endpoints, a single named palette, or a palette function used for the quantile groups.
quantiles	The number of quantile groups.
oob	Out-of-bounds handling function or shortcut. Use a function like scales::oob_squish() or a shortcut such as "censor" or "squish".
right	Whether intervals should be closed on the right. The default of FALSE yields intervals like [a, b).
labels	An optional labeling function or a character vector for the quantile ranges. When a function is supplied, it is applied to the quantile boundaries before interval labels are constructed.
title	Optional legend title.
width	Width of the legend.
height	Height of the swatches.
apply_to	Whether colors should be applied to cell fill or text.
na_color	Color used for missing values.
alpha	Alpha applied by gt::data_color().
reverse	Whether to reverse the color mapping.
autocolor_text	Whether to automatically adjust text color.
contrast_algo	Contrast algorithm passed to gt::data_color().
autocolor_light	Light text color used by gt::data_color().
autocolor_dark	Dark text color used by gt::data_color().

Value

A modified gt::gt table.

Examples

```
library(gt)

exibble |>
  gt() |>
  gtscale_data_color_quantiles(
    column = num,
    palette = c('#fdd49e', '#fdbb84', '#ef6548', '#990000'),
    quantiles = 4,
    title = 'Quantile bins'
  )
```

---

gtscale_legend *Attach only the legend component of a* gtscales *spec*

Description

Attach only the legend component of a gtscales spec

Usage

```
gtscale_legend(data, spec)
```

Arguments

data	A gt_tbl created by gt::gt().
spec	A gtscale_spec.

Value

A modified gt::gt table.

---

gtscale_render_legend *Render a legend from a* gtscale_spec

Description

Render a legend from a gtscale_spec

Usage

```
gtscale_render_legend(
  spec,
  data = NULL,
  output = c("contextual", "html", "latex", "rtf", "word", "typst")
)
```

Arguments

spec	A gtscale_spec.
data	An optional gt_tbl used to finalize specs that infer domains or quantile boundaries from table data.
output	Output target. Use "html", "latex", "rtf", "word", "typst", or "contextual".

Value

Rendered legend content for the requested output target.

Examples

```
spec <- gtscale_spec_quantiles(
  num,
  palette = c('#fdd49e', '#fdbb84', '#ef6548', '#990000'),
  quantiles = 4,
  title = 'Quartiles'
)

gtscale_render_legend(
  spec = spec,
  data = gt::gt(gt::exibble),
  output = 'latex'
)
```

---

gtscale_spec_bins *Create a binned* gtscales *spec*

Description

Create a binned gtscales spec

Usage

```
gtscale_spec_bins(
  column,
  palette,
  bins = NULL,
  domain = NULL,
  transform = NULL,
  oob = NULL,
  right = FALSE,
  labels = NULL,
  title = NULL,
  width = "180px",
  height = "14px"
)
```

Arguments

column	A column to target.
palette	A vector of colors or palette endpoints used for the bins. A single named palette or palette function can also be supplied.
bins	Optional bin boundaries or a break function. When omitted, breaks are generated from the domain, column, and transform.
domain	Optional limits. If omitted, these can be inferred when the spec is applied to a gt table.
transform	A transformation specification understood by scales::as.transform(). This is used when generating default bins or when interpreting break functions.
oob	Out-of-bounds handling function or shortcut. Use a function like scales::oob_squish() or a shortcut such as "censor" or "squish".
right	Whether intervals should be closed on the right. The default of FALSE yields intervals like [a, b).
labels	An optional labeling function or character vector for the legend. When omitted, labels are inferred from the bin values.
title	Optional legend title.
width	Width of the legend.
height	Height of the swatches.

Value

A gtscale_spec.

---

gtscale_spec_continuous

Create a continuous gtscales *spec*

Description

Create a continuous gtscales spec

Usage

```
gtscale_spec_continuous(
  column,
  palette = NULL,
  domain = NULL,
  breaks = NULL,
  labels = NULL,
  title = NULL,
  transform = NULL,
  oob = NULL,
```

```
  direction = "to right",
  width = "160px",
  height = "14px",
  fn = NULL
)
```

Arguments

column	A column to target.
palette	A vector of colors used in the scale. A single named palette such as "viridis" or "Blues 3", or a palette function, can also be supplied.
domain	Optional numeric limits. If omitted, these can be inferred when the spec is applied to a gt table.
breaks	Optional break values or a break function for the legend.
labels	An optional labeling function or character vector for the legend. When omitted, labels are inferred from the data or transform.
title	Optional legend title.
transform	A transformation specification understood by scales::as.transform(). When omitted, an appropriate identity, date, time, or timespan transform is inferred from the data.
oob	Out-of-bounds handling function or shortcut passed through to the internal color mapper. Use a function like scales::oob_squish() or a shortcut such as "censor", "squish", "keep", or "discard".
direction	CSS gradient direction. Defaults to "to right".
width	Width of the legend bar.
height	Height of the legend bar.
fn	Optional scales function for numeric coloring.

Value

A gtscale_spec.

---

gtscale_spec_discrete *Create a discrete* gtscales *spec*

Description

Create a discrete gtscales spec

Usage

```
gtscale_spec_discrete(
  column,
  values,
  labels = values,
  title = NULL,
  swatch_size = "12px",
  levels = NULL,
  ordered = FALSE
)
```

Arguments

column	A column to target.
values	A vector of color values, a single named discrete palette, or a discrete palette function.
labels	Labels for each legend swatch. Defaults to values.
title	Optional legend title.
swatch_size	Size of each discrete color swatch.
levels	Optional factor levels.
ordered	Whether the scale should be treated as ordered.

Value

A gtscale_spec.

---

gtscale_spec_diverging

Create a diverging gtscales *spec*

Description

Create a diverging gtscales spec

Usage

```
gtscale_spec_diverging(
  column,
  palette,
  domain = NULL,
  midpoint = 0,
  breaks = NULL,
  labels = NULL,
  title = NULL,
  transform = NULL,
```

```
  oob = NULL,
  direction = "to right",
  width = "160px",
  height = "14px",
  mid_color = "#FFFFFF"
)
```

Arguments

column	A column or shared set of columns to target.
palette	Two endpoint colors or three diverging colors. A single named palette such as "Blue-Red 3" or "viridis", or a palette function, can also be supplied.
domain	Optional numeric limits. If omitted, these can be inferred when the spec is applied to a gt table.
midpoint	Numeric midpoint used to anchor the diverging scale.
breaks	Optional break values or a break function for the legend.
labels	An optional labeling function or character vector for the legend. When omitted, labels are inferred from the data or transform.
title	Optional legend title.
transform	A transformation specification understood by scales::as.transform(). When omitted, an appropriate identity, date, time, or timespan transform is inferred from the data.
oob	Out-of-bounds handling function or shortcut passed through to the internal color mapper. Use a function like scales::oob_squish() or a shortcut such as "censor", "squish", "keep", or "discard".
direction	CSS gradient direction. Defaults to "to right".
width	Width of the legend bar.
height	Height of the legend bar.
mid_color	Midpoint color when palette supplies only two endpoint colors.

Value

A gtscale_spec.

---

gtscale_spec_quantiles

Create a quantile gtscales *spec*

Description

Create a quantile gtscales spec

Usage

```
gtscale_spec_quantiles(
  column,
  palette,
  quantiles = 4,
  oob = NULL,
  right = FALSE,
  labels = NULL,
  title = NULL,
  width = "180px",
  height = "14px"
)
```

Arguments

column	A numeric, Date, POSIXt, or difftime column to target.
palette	A vector of colors or palette endpoints used for the quantile groups. A single named palette or palette function can also be supplied.
quantiles	The number of quantile groups.
oob	Out-of-bounds handling function or shortcut. Use a function like scales::oob_squish() or a shortcut such as "censor" or "squish".
right	Whether intervals should be closed on the right. The default of FALSE yields intervals like [a, b).
labels	An optional labeling function or character vector for the legend. When omitted, labels are inferred from the quantile break values.
title	Optional legend title.
width	Width of the legend.
height	Height of the swatches.

Value

A gtscale_spec.

---

gtscale_spec_set_application

Set how a gtscales *spec is applied*

Description

Set how a gtscales spec is applied

Usage

```
gtscale_spec_set_application(
  spec,
  apply_to = c("fill", "text"),
  na_color = NULL,
  alpha = NULL,
  reverse = FALSE,
  autocolor_text = TRUE,
  contrast_algo = c("apca", "wcag"),
  autocolor_light = "#FFFFFF",
  autocolor_dark = "#000000"
)
```

Arguments

spec	A gtscale_spec.
apply_to	Whether colors should be applied to cell fill or text.
na_color	Color used for missing values.
alpha	Alpha applied by gt::data_color().
reverse	Whether to reverse the color mapping.
autocolor_text	Whether to automatically adjust text color.
contrast_algo	Contrast algorithm passed to gt::data_color().
autocolor_light	Light text color used by gt::data_color().
autocolor_dark	Dark text color used by gt::data_color().

Value

A modified gtscale_spec.

---

gtscale_spec_set_legend

Set how a gtscales *legend should be rendered*

Description

Set how a gtscales legend should be rendered

Usage

```
gtscale_spec_set_legend(
  spec,
  output = "contextual",
  placement = "source_note",
  layout = c("stack", "inline"),
  align = c("left", "center", "right"),
  show_border = TRUE,
  border_color = "#D0D7DE",
  border_radius = "8px",
  show_na = FALSE,
  na_label = "Missing",
  na_color = NULL
)
```

Arguments

spec	A gtscale_spec.
output	Output target for the legend. Use "contextual" for gt-managed HTML/LaTeX source notes, or choose a specific output like "html", "latex", or "typst".
placement	Legend placement target. "source_note", "title", and "subtitle" are currently implemented.
layout	Whether multiple legends in the same heading area should stack vertically or sit inline.
align	Horizontal alignment for the legend container.
show_border	Whether the legend bar, bin frame, and swatches should draw borders.
border_color	Border color used for legend frames and swatches.
border_radius	Border radius used for HTML and Typst legend frames.
show_na	Whether to include an explicit missing-value legend entry.
na_label	Label to use for missing values in the legend.
na_color	Optional legend swatch color for missing values. When omitted and show_na = TRUE, a neutral gray swatch is used.

Value

A modified gtscale_spec.

Index