Package ‘ggfourfold’

October 21, 2026

Type Package

Title Fourfold Displays for 'ggplot2'

Version 0.1.0

Language en-US

Maintainer Gavin Klorfine <gklorfin@yorku.ca>

Depends R (>= 4.1.0)

Imports ggplot2 (>= 4.0.0), grDevices, grid, stats

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

Description Draws fourfold displays of 2 x 2 (x k) contingency tables as a 'ggplot2' layer, following Friendly (1994) <https://datavis.ca/papers/4fold/4fold.pdf> and Friendly and Meyer (2016) <doi:10.1201/9781315374413>. Cell frequencies are shown as quarter-circles or squares of proportional area, with confidence rings for the odds ratio and shading for the direction and strength of association. Stratified tables are shown with 'ggplot2' faceting and a corresponding 'ggplot2' theme is provided.

License GPL (>= 2)

Encoding UTF-8

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

VignetteBuilder knitr

URL https://gavinklorfine.com/ggfourfold/, https://github.com/gklorfine/ggfourfold

BugReports https://github.com/gklorfine/ggfourfold/issues

Collate 'fourfold-palette.R' 'geom-fourfold.R' 'theme-fourfold.R'

NeedsCompilation no

Author Gavin Klorfine [aut, cre, cph] (ORCID: <https://orcid.org/0009-0003-0395-4512>), Michael Friendly [aut] (ORCID: <https://orcid.org/0000-0002-3237-0941>)

Repository CRAN

Date/Publication 2026-10-21 14:20:11 UTC

Contents

fourfold_palette 2
geom_fourfold 3
theme_fourfold 10

Index **12**

`fourfold_palette` *Palettes for fourfold displays*

Description

`fourfold_palette()` returns six colors for the `palette` argument of `geom_fourfold()`. The colors encode the direction and statistical strength of association and are drawn directly by the geom rather than through a fill scale. Entries 1-2 are used when `extended = FALSE` (or `conf_level = 0`), entries 3-4 are used for a stratum whose association *is not* significant after adjustment, and entries 5-6 are used for a stratum whose association *is* significant after adjustment. Within each pair, the first color fills the diagonal with fewer cases than expected under independence and the second fills the diagonal with more.

Usage

```
fourfold_palette(palette = c("vcd", "okabe-ito"))
```

Arguments

`palette`	Name of the palette: `"vcd"` (the default) or `"okabe-ito"`.

Details

`"vcd"`, the default, gives the colors of `vcd::fourfold()` (Meyer et al., 2026): light and darker blue, then light red and light blue, then red and navy.

`"okabe-ito"` uses the colors of Okabe and Ito (2008), chosen to remain distinguishable for readers with common forms of color-vision deficiency: sky blue and blue, then orange and sky blue lightened about halfway to white, then vermilion and blue.

Value

A character vector containing six hexadecimal colors.

References

Meyer, D., Zeileis, A., Hornik, K., & Friendly, M. (2026). *vcd: Visualizing Categorical Data* (R package). doi:10.32614/CRAN.package.vcd

Okabe, M., & Ito, K. (2008). *Color Universal Design (CUD): How to make figures and presentations that are friendly to colorblind people*. https://jfly.uni-koeln.de/color/

See Also

geom_fourfold() and grDevices::palette.colors()

Examples

```
fourfold_palette()
fourfold_palette("okabe-ito")

ucb <- as.data.frame(UCBAdmissions)

ggplot2::ggplot(
  ucb,
  ggplot2::aes(x = Gender, y = Admit, weight = Freq)
) +
  geom_fourfold(palette = fourfold_palette("okabe-ito")) +
  ggplot2::facet_wrap(ggplot2::vars(Dept), ncol = 3) +
  theme_fourfold()
```

---

geom_fourfold *Fourfold displays for 2-by-2 tables*

Description

geom_fourfold() draws a fourfold display in each **ggplot2** panel. Sector areas are proportional to cell frequencies after the selected standardization, while sector colors, confidence rings, and a diagonal line show the direction and strength of association.

Usage

```
geom_fourfold(
  mapping = NULL,
  data = NULL,
  ...,
  std = c("margins", "ind.max", "all.max"),
  margin = c(1, 2),
  conf_level = 0.95,
  extended = TRUE,
  diagonal = TRUE,
  diagonal.length = NULL,
  diagonal.fill = "white",
  diagonal.width = NULL,
  p_adjust_method = stats::p.adjust.methods,
  shape = c("circle", "square"),
  counts = c("auto", "inside", "outside", "none"),
  palette = fourfold_palette(),
  na.rm = FALSE,
  show.legend = FALSE,
```

```
  inherit.aes = TRUE
)
```

Arguments

`mapping` Set of aesthetic mappings created by `ggplot2::aes()`. If supplied and `inherit.aes = TRUE`, these are combined with the plot's default mappings. Defaults to `NULL`.

`data` The data to display in this layer. If `NULL`, the default, the data are inherited from the plot.

`...` Other arguments passed to `ggplot2::layer()`, typically fixed aesthetics such as `color` or `linewidth`.

`std` `"margins"` (the default) draws a standardized display, equating the margins chosen by `margin` while preserving the odds ratio. `"ind.max"` and `"all.max"` draw unstandardized displays of the raw counts, respectively scaled by the largest cell in each panel or in the whole layer.

`margin` Numeric vector selecting the table margins when `std = "margins"`. Use `c(1, 2)` (the default) for both margins, `1` for the `y` (row) margin, or `2` for the `x` (column) margin, as in `vcd::fourfold()`.

`conf_level` Confidence level in `[0, 1)`. Defaults to `0.95`; set to `0` to suppress confidence rings and significance shading.

`extended` A single `TRUE` (the default) or `FALSE`. If `FALSE`, omit the diagonal line, which marks the diagonal with more cases than expected under independence, and fill cells with the first two `palette` colors instead of shading by significance.

`diagonal` A single `TRUE` (the default) or `FALSE`: draw the diagonal line of an extended display (see `extended`) or not.

`diagonal.length` How far each end of the diagonal line extends past the outline of the sector it marks, measured along the diagonal for both shapes (past the arc of a circle, or the outer corner of a square), where `1` is the largest possible sector radius. If `NULL`, the default, it is `0.15` for circles and `0.02 * sqrt(2)` (about `0.028`) for squares, which is `0.02` on each axis; a number applies to either shape as given. With `0`, the line ends at the sectors but is still drawn: use `diagonal = FALSE` to omit it. Supply `NULL` or a single non-negative number.

`diagonal.fill` The color of the diagonal line's interior (its border is drawn in the layer's `color`). Defaults to `"white"`. Like `fill` in **ggplot2**, `NA` or `"transparent"` leaves the line hollow, with its border only. Supply a single color.

`diagonal.width` The width of the diagonal line's interior, in millimeters like `linewidth`. This width does not count the line's border, which is drawn outside it with the layer's `color` and `linewidth`. If `NULL`, the default, it is 2.5 times the layer's `linewidth`. Supply `NULL` or a single non-negative number. With `0`, the two borders meet and the line interior disappears, leaving a line in the layer's `color` that is twice the layer's `linewidth` wide.

`p_adjust_method` Method passed to `stats::p.adjust()` for adjustment across panels. Defaults to `"holm"`, the first value in stats::p.adjust.methods.

shape	Shape of the cell sectors. "circle" (the default, as in vcd::fourfold()) draws quarter-circles. "square" draws squares with equivalent area to the corresponding quarter-circle.
counts	Placement of cell counts: "auto" (the default) moves counts outside when the drawing approaches them; "inside" always uses the inside corners, even if overlapped; "outside" always uses the outside corners; "none" hides the counts. Counts are labeled as in vcd::fourfold(), so large round counts stored as doubles can appear as, for example, 1e+05. Non-integer counts are rounded to three significant digits, and rounding noise such as 5.6e-17 is shown as 0.
palette	Character vector of at least six valid colors in the semantic order used by fourfold_palette(). Defaults to fourfold_palette(), the colors of vcd::fourfold(). See fourfold_palette() for other built-in palettes such as fourfold_palette("okabe-ito").
na.rm	Missing values are handled the same way either way. Rows with NA in x or y are removed, and a panel with an NA weight is left empty. If FALSE, the default, a warning says so. If TRUE, this is done silently. See the Missing values section.
show.legend	Logical indicating whether this layer should be included in legends. The default is FALSE because the semantic fills are not a mapped aesthetic.
inherit.aes	If FALSE, override rather than combine with the plot's default aesthetic mappings. Defaults to TRUE.

Details

Map the binary horizontal variable to x, the binary vertical variable to y, and cell frequencies to weight. When weight is omitted, each row counts as one observation. The first x level is drawn on the left and the second on the right; the first y level is drawn at the top and the second at the bottom (see the Coordinate systems section). Set factor levels explicitly when their order matters. Alternatively, reorder categories with the limits argument of ggplot2::scale_x_discrete() or ggplot2::scale_y_discrete(), and rename them with labels. Scale breaks that reorder or omit categories, and a scale palette that moves them, are errors.

One panel must contain exactly one 2-by-2 table. Use ggplot2::facet_grid() or ggplot2::facet_wrap() to display stratified tables. Duplicate x/y combinations within a panel are summed and cells with no rows are completed with zero counts. For missing values, see the Missing values section.

Odds ratios, Wald confidence intervals, and extended-display p-values match the calculations in vcd::fourfold(). If any observed cell is zero, 0.5 is added to all four cells for inference; see the Zero counts section for how such tables are drawn. p-values are adjusted for multiple testing across the panels of the layer. A panel whose counts are all zero has no p-value and is left out of this adjustment. vcd::fourfold() instead includes such a panel with a p-value of 1, so in a layer that contains one, the adjusted p-values of the other panels are smaller here than in vcd::fourfold(). Confidence intervals themselves are not adjusted.

With shape = "square", each cell is drawn as a square with the same area as the corresponding quarter-circle (side $r\sqrt{\pi}/2$ for radius r), so the two shapes display a table with identical areas. Confidence rings become nested square outlines.

In an extended display (the default), one straight diagonal line through the center of the display marks the diagonal with more cases than expected under independence–the direction of the association. The line is a band, drawn like a rectangle in ggplot2::geom_rect(), with a border along both sides and across both ends so that it stands out on pale and dark sectors alike. By default

its interior is white and 2.5 times as wide as the layer's `linewidth`. Use `diagonal.fill` and `diagonal.width` to change these. The border is drawn in the layer's `color` and `linewidth`. Each end extends `diagonal.length` past the sector it marks, measured along the diagonal in units of half the frame's width. By default, this length is 0.15 units past the arc of a circle, or `0.02 * sqrt(2)` (0.02 units on each axis) units past the outer corner of a square, short enough that the counts of a typical standardized display stay inside. The line can be omitted by setting `diagonal = FALSE`. This differs from `vcd::fourfold()`, which marks the direction with two short black ticks, one at each of the two sectors on that diagonal, instead of a line through the center. **vcd**'s `ticks` argument, which controls the length of those ticks, corresponds to `diagonal.length`, except that `diagonal.length = 0` still draws the line (ending at the sectors), whereas `ticks = 0` in `vcd::fourfold()` draws no ticks.

With `counts = "auto"`, cell counts stay inside the frame corners unless a sector, either confidence outline, or an end of the diagonal line comes close to the count text. If they come close, counts then move just outside the frame corners. Clearance is measured at draw time, so placement can differ between panels of different physical sizes. Use `counts = "inside"` or `"outside"` to force the placement, or `"none"` to hide counts. Outside counts can collide with category labels or neighboring panels at small sizes. If this occurs, use larger panels, more panel spacing, smaller text, or `counts = "inside"`.

The six semantic fill colors are supplied by `palette`; they are not mapped through a **ggplot2** fill scale. Typography and layout defaults are controlled by `theme_fourfold()`.

Value

A list of a **ggplot2** layer and a default coordinate system, `coord_cartesian(reverse = "y", ratio = 1)`, which can be added to a `ggplot2::ggplot()` object (see the Coordinate systems section). `GeomFourfold` and `StatFourfold` are the `ggplot2::ggproto()` objects behind `geom_fourfold()`, a `Geom` and a `Stat`, for use with `ggplot2::layer()`.

Aesthetics

`geom_fourfold()` understands the following aesthetics:

- `x` (required): a categorical variable (factor, character, or logical) with exactly two levels. Convert numeric codes, such as 0/1, with `factor()`.
- `y` (required): a categorical variable with exactly two levels.
- `weight`: non-negative numeric cell frequencies; defaults to `1`.

Give `color`, `linewidth`, and `alpha` as fixed arguments rather than mappings, for example `geom_fourfold(color = "grey30")`. As in **ggplot2**'s filled geoms, `alpha` sets the transparency of the cell fills only. Fill colors are set with `palette`. Text `size` and `family` come from the plot theme, such as `theme_fourfold()`. To override them for this layer only, give them to `geom_fourfold()`, for example `geom_fourfold(family = "serif")`. As elsewhere in **ggplot2**, `size` is in mm, while the theme's `base_size` is in points.

Mapping a `color`, `linewidth`, `alpha`, `size`, or `family` aesthetic in `geom_fourfold()` is an error. This includes mappings to computed variables with `ggplot2::after_stat()`, `ggplot2::after_scale()`, or `ggplot2::stage()`. The same mappings inherited from `ggplot2::ggplot()` are ignored, because they may be meant for other layers. The one exception is an inherited `after_stat()` mapping to a variable that `geom_fourfold()` does not compute, such as `after_stat(n)` for `ggplot2::geom_count()`.

ggplot2 evaluates it before the layer can ignore it, so the plot stops with an error. To avoid this, set `inherit.aes = FALSE` in `geom_fourfold()`, or move the mapping into the layer that uses it.

Coordinate systems

The display is drawn in the plot's own coordinates, so axes, gridlines, and other layers line up with it. Each category sits at its usual discrete position, with the first `x` level at 1 and the second at 2, and likewise for `y`. The frame spans 0.5 to 2.5 on both axes, so each cell's quadrant is centered on its two category positions. A label placed at `x = 1, y = 2`, for example, lands in the middle of the cell for the first level of `x` and the second level of `y`.

`geom_fourfold()` therefore also adds `coord_cartesian(reverse = "y", ratio = 1)` to the plot, as `ggplot2::geom_sf()` adds `ggplot2::coord_sf()`. `reverse = "y"` keeps the first `y` level at the top, as in `vcd::fourfold()`, for every layer. `ratio = 1` keeps circles round, and because both axes span the same range by default, it also makes the panels square. A theme's `aspect.ratio` overrides `ratio`, fixing the panel's shape, with circles turning into ellipses when the ranges of the axes differ (see the Missing values section).

A coordinate system that you add to the plot replaces this one, as usual in **ggplot2**. Add yours after the last `geom_fourfold()` in the plot, because a later `geom_fourfold()` replaces it with its own. Without `reverse = "y"`, the first `y` level is drawn at the bottom, as for any **ggplot2** layer. To keep it at the top and the circles round, add `coord_cartesian(reverse = "y", ratio = 1, ...)`. Used directly with `ggplot2::layer()`, `GeomFourfold` and `StatFourfold` take the defaults of `geom_fourfold()` but add no coordinate system.

The coordinate system must be Cartesian, such as `ggplot2::coord_cartesian()` or `ggplot2::coord_fixed()`. `ggplot2::coord_flip()` is an error: swap the `x` and `y` aesthetics instead, which transposes each table and keeps its odds ratio. So are `ggplot2::coord_sf()`, `ggplot2::coord_transform()`, and polar coordinates, which would distort the areas that carry the display's meaning. Because of the fixed `ratio`, **ggplot2** does not allow free facet scales (`scales = "free"`) with the default coordinate system. To use them, add your own coordinate system without `ratio` after `geom_fourfold()`, such as `coord_cartesian(reverse = "y")`. Then `facet_grid(space = "free")` also works, including with `theme_fourfold()`, which sets no aspect ratio; it shares widths among columns (heights among rows) in proportion to the ranges of their axes. Free scales and free space are therefore possible only with your own coordinate system without `ratio`, and the displays are then round only where a panel happens to be square.

The display is not clipped to the panel, so that counts outside the frame stay whole. Zooming with the `xlim` and `ylim` of the coordinate system therefore does not crop it, and the display can then extend over neighboring panels and strips.

Annotations

A point or label that another layer, such as `ggplot2::geom_point()`, `ggplot2::geom_text()`, or `ggplot2::annotate()`, places at an `x` level and a `y` level is drawn at the center of that cell's quadrant. This position is fixed, the same in every panel, and does not follow the cell's filled quarter-circle or square, whose size varies with the data. See the examples.

Zero counts

If any cell of a panel's table is zero, 0.5 is added to all four cells before the odds ratio, its standard error, and the confidence interval are computed, as in `vcd::fourfold()`. The count labels always

show the observed counts.

A panel whose four counts are all zero, such as an empty stratum in a faceted display, is drawn blank: only its frame, axes, labels, and zero counts are shown, with no sectors, rings, or diagonal line. It has no odds ratio, confidence interval, or *p*-value (they are `NA`) and is left out of the *p*-value adjustment, so it does not change the other panels. With the default `margin = c(1, 2)`, `vcd::fourfold()` instead draws such a stratum from the corrected table, as four equal quarter-circles. A facet level with no rows at all, for example with `drop = FALSE`, is left empty by **ggplot2** as usual, without a frame or counts.

Each confidence ring shows the table that has the observed row and column totals and an odds ratio equal to one confidence limit, standardized like the data. With the default `margin = c(1, 2)`, which depends only on the odds ratio, this is the fully standardized table at that limit. With the other `std` and `margin` settings, if a whole row or column is zero, the observed totals admit no other table, so the rings of that panel are built from the row and column totals of the table with 0.5 added to every cell, rescaled to the observed total (which matters only for `std = "all.max"`). Panels with isolated zero cells, but no empty row or column, keep the observed totals for their rings.

A table with an empty row or column contains no information about the odds ratio: with an empty row, for example, nothing is known about how that row would split between the columns. Its odds ratio and *p*-value exist only because of the correction. With an empty row, the odds ratio is the ratio of the two counts in the other row, each plus 0.5, so it reflects how that row splits rather than an association. The rings are correspondingly wide, but the *p*-value can still fall below the significance level, so the color, significance shading, and diagonal line of such a panel say nothing about an association.

The sectors show the observed table wherever the standardization can use it:

- With `std = "margins"` and the default `margin = c(1, 2)`, the sectors depend only on the odds ratio, which already includes the correction.
- With `margin = 1`, a row that is entirely zero has no proportions, so it is drawn from the corrected table, as two equal quarter-circles. The other row keeps its observed proportions. The same applies to an empty column with `margin = 2`. An empty column with `margin = 1`, or an empty row with `margin = 2`, keeps its observed proportions and so has no sectors, except in a cell it shares with an empty row (or column) that is drawn from the corrected table.
- With `std = "ind.max"` or `"all.max"`, the sectors show the observed counts, so an empty row or column has no sectors. Its rings still come from the corrected totals, so they need not match the sectors.

Weights that are very small but not zero, such as the remainder of floating-point arithmetic (`0.1 + 0.2 - 0.3`), count as observations, so a row or column of them is not empty and its rings keep the observed totals. Where the standardization keeps such a row or column near zero (`std = "ind.max"` or `"all.max"`, `margin = 2` for a row, or `margin = 1` for a column), its near-zero totals leave almost no room for other odds ratios, so the rings lie on the sectors, as if the estimate were precise. An exactly empty row or column gets the wide rings described above instead. Round such weights, for example with `round(w, 8)`, if they are meant to be zero.

Missing values

A row with a missing `x` or `y` cannot be placed in the table, so it is removed, and a panel with no rows left is left empty. Values excluded by the `limits` of a discrete scale become missing and are removed the same way. A row with a missing `weight` but known `x` and `y` is different. Its cell's

count, and so the panel's table, is unknown, and counting it as zero would change the odds ratio. Its whole panel is therefore left empty. A cell with no rows at all is not missing but a zero count. With a missing count, `vcd::fourfold()` instead stops with an error.

An empty panel has no frame, counts, or labels, so it cannot be mistaken for a panel whose counts are all zero, and it looks the same as a facet level with no rows. It has no rows in the layer data and takes no part in `std = "all.max"` or the *p*-value adjustment, so the other panels are drawn exactly as if its stratum were not in the data. With `na.rm = FALSE`, the default, a warning names each panel that lost rows or was left empty. With `na.rm = TRUE`, these warnings are not given. With free scales, which need a coordinate system added without `ratio` (see the Coordinate systems section), **ggplot2** itself may still warn about the axes of an empty panel ("Position guide is perpendicular to the intended axis"), as it does for its own layers.

`NA` values in the `x` or `y` variable leave empty space beside the display, which can then be drawn smaller. Only the layout changes: panels that share the scale shrink alike and stay comparable, the display stays round with the default coordinate system, and the areas, counts, and statistics are not affected. To remove the space, add `scale_x_discrete(na.translate = FALSE)` for `x`, `scale_y_discrete(na.translate = FALSE)` for `y`, or both. The space appears because **ggplot2** trains position scales before the stat runs, so the axis keeps a place for missing values, as for every geom, even though the display removes those rows.

References

Friendly, M. (1994a). *A fourfold display for 2 by 2 by k tables* (Technical Report No. 217). York University, Psychology Department. https://datavis.ca/papers/4fold/4fold.pdf

Friendly, M. (1994b). SAS/IML graphics for fourfold displays. *Observations*, *3*(4), 47–56.

Friendly, M., & Meyer, D. (2016). *Discrete Data Analysis with R: Visualization and Modeling Techniques for Categorical and Count Data* (Section 4.4). Chapman & Hall/CRC. http://ddar.datavis.ca

Meyer, D., Zeileis, A., Hornik, K., & Friendly, M. (2026). *vcd: Visualizing Categorical Data* (R package). doi:10.32614/CRAN.package.vcd

See Also

`theme_fourfold()`, `fourfold_palette()`, `ggplot2::facet_grid()`, and `ggplot2::facet_wrap()`

Examples

```
ucb <- as.data.frame(UCBAdmissions)

ggplot2::ggplot(
  ucb,
  ggplot2::aes(x = Gender, y = Admit, weight = Freq)
) +
  geom_fourfold() +
  ggplot2::facet_wrap(ggplot2::vars(Dept), ncol = 3) +
  ggplot2::labs(title = "Berkeley admissions") +
  theme_fourfold()

# Squares instead of quarter-circles
ggplot2::ggplot(
```

```
  ucb,
  ggplot2::aes(x = Gender, y = Admit, weight = Freq)
) +
  geom_fourfold(shape = "square") +
  ggplot2::facet_wrap(ggplot2::vars(Dept), ncol = 3) +
  ggplot2::labs(title = "Berkeley admissions") +
  theme_fourfold()

# A longer diagonal line, in light gray
ggplot2::ggplot(
  ucb,
  ggplot2::aes(x = Gender, y = Admit, weight = Freq)
) +
  geom_fourfold(diagonal.length = 0.3, diagonal.fill = "gray80") +
  ggplot2::facet_wrap(ggplot2::vars(Dept), ncol = 3) +
  ggplot2::labs(title = "Berkeley admissions") +
  theme_fourfold()

# Other layers are placed by category: admission rates in the admitted cells
ggplot2::ggplot(
  subset(ucb, Dept == "A"),
  ggplot2::aes(x = Gender, y = Admit, weight = Freq)
) +
  geom_fourfold() +
  ggplot2::annotate(
    "label", x = c("Male", "Female"), y = "Admitted",
    label = c("62% admitted", "82% admitted")
  ) +
  theme_fourfold()
```

`theme_fourfold` *Theme for fourfold displays*

Description

`theme_fourfold()` is a minimal theme for `geom_fourfold()` that gives plots the plain look of `vcd::fourfold()`: a white panel without axes or gridlines, facet labels without boxes, centered titles, and compact spacing between panels. It also sets the text size, font, and line width.

Usage

```
theme_fourfold(base_size = 12, base_family = "", ...)
```

Arguments

`base_size`	Base font size in points.
`base_family`	Base font family. The default, `""`, uses the graphics device's default family.
`...`	Additional arguments passed to `ggplot2::theme()`. They are applied after the fourfold defaults.

Details

base_size and base_family control all text, including the category labels and cell counts drawn by geom_fourfold(). These grow with the size of each panel, so larger plots get larger labels, but never shrink below five-sixths of base_size (10 points by default). Additional theme elements passed through ... are applied last and therefore override the defaults.

The theme sets no aspect.ratio. The display stays round, and the panels square by default, because geom_fourfold() adds a coordinate system with ratio = 1 (see the Coordinate systems section of geom_fourfold()). Passing aspect.ratio through ... fixes the panel's shape and overrides that ratio, so circles become ellipses when the ranges of the x and y axes differ, for example when a missing value keeps a place on one axis. Therefore, it is recommended to leave aspect.ratio unset.

This theme uses **ggplot2**'s theme-derived geom defaults and requires **ggplot2** 4.0.0 or later.

Value

A complete **ggplot2** theme.

See Also

geom_fourfold() and ggplot2::theme()

Examples

```
ucb <- as.data.frame(UCBAdmissions)

ggplot2::ggplot(
  ucb,
  ggplot2::aes(x = Gender, y = Admit, weight = Freq)
) +
  geom_fourfold() +
  ggplot2::facet_wrap(ggplot2::vars(Dept), ncol = 3) +
  theme_fourfold(base_size = 12)
```

Index

fourfold_palette, 2
fourfold_palette(), *5*, *9*

geom_fourfold, 3
geom_fourfold(), *2*, *3*, *10*, *11*
GeomFourfold *(*geom_fourfold*)*, 3
ggplot2::aes(), *4*
ggplot2::after_scale(), *6*
ggplot2::after_stat(), *6*
ggplot2::annotate(), *7*
ggplot2::coord_cartesian(), *7*
ggplot2::coord_fixed(), *7*
ggplot2::coord_flip(), *7*
ggplot2::coord_sf(), *7*
ggplot2::coord_transform(), *7*
ggplot2::facet_grid(), *5*, *9*
ggplot2::facet_wrap(), *5*, *9*
ggplot2::geom_count(), *6*
ggplot2::geom_point(), *7*
ggplot2::geom_rect(), *5*
ggplot2::geom_sf(), *7*
ggplot2::geom_text(), *7*
ggplot2::ggplot(), *6*
ggplot2::ggproto(), *6*
ggplot2::layer(), *4*, *6*, *7*
ggplot2::scale_x_discrete(), *5*
ggplot2::scale_y_discrete(), *5*
ggplot2::stage(), *6*
ggplot2::theme(), *10*, *11*
grDevices::palette.colors(), *3*

StatFourfold *(*geom_fourfold*)*, 3
stats::p.adjust(), *4*
stats::p.adjust.methods, *4*

theme_fourfold, 10
theme_fourfold(), *6*, *7*, *9*

vcd::fourfold(), *4*