Package: ggpyramid (via r-universe)

November 10, 2026

Title Population Pyramids with 'ggplot2'

Version 0.1.0

Description Build population pyramids, the back-to-back bar charts of a population by age group and sex, as layers of a 'ggplot2' plot. A position adjustment mirrors one side and stacks subgroups within each side, so bars, labels and lines from any layer land on the correct side while the plot can still be faceted, themed and extended as usual. A coordinate system labels both sides with absolute values, gives them one shared scale, and places the age axis on either side or in a gap at the centre. Further layers count individual records or sum aggregated values, show proportions, label the bars, and draw a second population as a stepped outline for comparison.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

Imports cli, ggplot2 (>= 4.0.0), rlang, scales, stats, utils, vctrs

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

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/matanhakim/ggpyramid, https://matanhakim.github.io/ggpyramid/

BugReports https://github.com/matanhakim/ggpyramid/issues

LazyData true

NeedsCompilation no

Author Matan Hakim [aut, cre, cph] (ORCID: <https://orcid.org/0009-0002-9372-8975>)

Maintainer Matan Hakim <matanhakim@gmail.com>

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

Date/Publication 2026-11-10 18:00:02 UTC

RemoteUrl https://github.com/cran/ggpyramid

RemoteRef HEAD

RemoteSha 5d85f15ef03d4f1b2e2866d4952f7baa8da27a48

Contents

coord_pyramid 2
europe_pop 4
geom_pyramid 5
geom_pyramid_outline 8
geom_pyramid_text 10
position_pyramid 13
scale_x_pyramid 14
theme_pyramid 16

Index **17**

`coord_pyramid` *Cartesian coordinates for a population pyramid*

Description

`coord_pyramid()` is a Cartesian coordinate system for horizontal pyramids drawn with `position_pyramid()`. It labels the value axis without the minus sign on the left side, extends the value axis equally far on both sides of zero, and places the age axis on the left, on the right, on both sides or in the centre of the pyramid, optionally in a gap between the two sides.

Usage

```
coord_pyramid(
  age_axis = c("left", "right", "both", "centre", "none"),
  left = NULL,
  gap = NULL,
  symmetric = TRUE,
  xlim = NULL,
  ylim = NULL,
  expand = TRUE,
  clip = "on"
)
```

Arguments

age_axis	Where to draw the age axis: "left", "right", "both", "centre" (between the two sides, see Age axis below) or "none". "center" is accepted as well.
left	The value of the side variable drawn on the left, such as "Female", for every layer drawn with position_pyramid(). A layer can override it with position_pyramid(left = ...). Defaults to the first level when the side variable is a factor, and to the first value in sorted order otherwise. The legend keeps its own order; set it with, for example, scale_fill_discrete(breaks = c("Female", "Male")).
gap	The width of an empty strip between the two sides, as a proportion of the panel width. Defaults to 0.12 when age_axis is "centre", which leaves room for the age labels, and to 0 otherwise.
symmetric	If TRUE, the value axis extends equally far on both sides of zero, so the two sides share one scale. Ignored when xlim is set.
xlim, ylim	Limits for the x and y axes.
expand	If TRUE, the default, adds a small expansion factor to the limits to ensure that data and axes don't overlap. If FALSE, limits are taken exactly from the data or xlim/ylim. Giving a logical vector will separately control the expansion for the four directions (top, left, bottom and right). The expand argument will be recycled to length 4 if necessary. Alternatively, can be a named logical vector to control a single direction, e.g. expand = c(bottom = FALSE).
clip	Should drawing be clipped to the extent of the plot panel? A setting of "on" (the default) means yes, and a setting of "off" means no. In most cases, the default of "on" should not be changed, as setting clip = "off" can cause unexpected results. It allows drawing of data points anywhere on the plot, including in the plot margins. If limits are set via xlim and ylim and some data points fall outside those limits, then those data points may show up in places such as the axes, the legend, the plot title, or the plot margins.

Value

A ggproto object of class CoordPyramid.

Value axis

Only the labels lose their sign. The data and the scale keep the signed values that position_pyramid() produces, so breaks, limits and reference lines on the left side are still given as negative numbers. Any labels given to ggplot2::scale_x_continuous() apply to the absolute values, for example labels = scales::label_percent(). With a gap, a break at zero is drawn on both inner edges, so a gap narrower than two zero labels makes them overlap. The value axis must be linear, and a secondary value axis keeps its signs.

Age axis

"left" and "right" move the age scale to that side together with its title, and "both" duplicates it on the right, as scale_y_discrete(sec.axis = dup_axis()) would. "centre" draws the age labels inside the panel, above the layers and in the middle of the gap, styled by the axis.text.y.left theme element. If the labels are wider than the gap, increase gap.

See Also

position_pyramid(), which mirrors the left side.

Examples

```
library(ggplot2)

pop <- data.frame(
  age = factor(rep(c("0-19", "20-39", "40-59", "60+"), 2)),
  sex = factor(rep(c("Male", "Female"), each = 4), c("Male", "Female")),
  n = c(220, 200, 160, 90, 210, 205, 170, 120)
)
p <- ggplot(pop, aes(x = n, y = age, fill = sex)) +
  geom_pyramid()

p + coord_pyramid()

# Women on the left
p + coord_pyramid(left = "Female")

# Age labels in a gap between the two sides
p + coord_pyramid(age_axis = "centre")

# Age labels on both sides, with a narrow gap
p + coord_pyramid(age_axis = "both", gap = 0.08)
```

---

europe_pop *Population of four European countries by age group and sex*

Description

The population on 1 January 2004 and 2024 of France, Ireland, Italy and Poland, by five-year age group and sex, from Eurostat. It is included for examples and has the shape most population data come in: one row per country, year, age group and sex.

Usage

```
europe_pop
```

Format

A data frame with 288 rows and 5 columns:

country Country, a factor.

year Year of the population count, 2004 or 2024.

age Age group, a factor from `"0-4"` to `"85+"` in ascending order.

sex Sex, a factor with levels `"Male"` and `"Female"`.

population Number of people.

Source

Eurostat, "Population on 1 January by age group and sex" (demo_pjangroup), https://ec.europa.eu/eurostat/databrowser/view/demo_pjangroup, retrieved 25 October 2026 and reused under the Creative Commons Attribution 4.0 licence. Some 2024 figures are provisional.

Examples

```
library(ggplot2)

ggplot(europe_pop, aes(x = population, y = age, fill = sex)) +
  geom_pyramid(denominator = "total") +
  scale_x_continuous(labels = scales::label_percent()) +
  coord_pyramid() +
  facet_grid(year ~ country)
```

---

geom_pyramid *Bars of a population pyramid*

Description

`geom_pyramid()` draws the bars of a population pyramid: one bar per age group on each side, extending away from a shared age axis. It counts rows, so it accepts individual records (`aes(y = age, fill = sex)`), and it sums a value when one is mapped, so it also accepts aggregated data (`aes(x = n, y = age, fill = sex)`). The bars are mirrored and stacked by `position_pyramid()`, and every other part of the plot (scales, facets, themes and further layers) comes from ggplot2 as usual.

Usage

```
geom_pyramid(
  mapping = NULL,
  data = NULL,
  stat = "pyramid",
  position = "pyramid",
  ...,
  denominator = NULL,
  just = 0.5,
  na.rm = FALSE,
  orientation = NA,
  show.legend = NA,
  inherit.aes = TRUE
)

stat_pyramid(
  mapping = NULL,
  data = NULL,
  geom = "bar",
```

```
  position = "pyramid",
  ...,
  denominator = NULL,
  na.rm = FALSE,
  orientation = NA,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
stat	The statistical transformation to use on the data for this layer. The default, "pyramid", is stat_pyramid().
position	A position adjustment. The default, "pyramid", is position_pyramid(); call it to set its arguments, for example position = position_pyramid(reverse = TRUE). To choose the side drawn on the left for every layer, use coord_pyramid(left = "Female").
...	Other arguments passed on to ggplot2::layer(), such as the bar width or fixed aesthetics like colour = "white".
denominator	NULL (the default) shows counts. "total" shows each bar as a proportion of all rows in the panel, and "side" as a proportion of the rows on its side.
just	Adjustment for column placement. Set to 0.5 by default, meaning that columns will be centered about axis breaks. Set to 0 or 1 to place columns to the left/right of axis breaks. Note that this argument may have unintended behaviour when used with alternative positions, e.g. position_dodge().
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
orientation	The orientation of the layer. The default (NA) automatically determines the orientation from the aesthetic mapping. In the rare event that this fails it can be given explicitly by setting orientation to either "x" or "y". See the *Orientation* section for more detail.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To

include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.

`inherit.aes` If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `annotation_borders()`.

`geom` The geometric object used to display the data.

Value

A ggplot2 layer.

Aesthetics

Age goes on `y`, or on `x` for a vertical pyramid. The value axis is optional: when it is mapped, its values are summed within each age group, and when it is not, rows are counted. `weight` multiplies either. The side of each bar comes from `side`, or from `fill` when `side` is not mapped; see `position_pyramid()`.

Computed variables

These can be used with `ggplot2::after_stat()`:

- `after_stat(count)`: the number of rows, or the sum of the value, in each age group. This is what the bars show by default.
- `after_stat(prop)`: `count` as a proportion of all rows in the panel, or of the rows on the same side with `denominator = "side"`. Each facet panel has its own denominator. The bars show `prop` whenever `denominator` is set.

See Also

`position_pyramid()` to mirror other layers, and `coord_pyramid()` for the axes of a pyramid.

Examples

```
library(ggplot2)

pop <- data.frame(
  age = factor(rep(c("0-19", "20-39", "40-59", "60+"), 2)),
  sex = factor(rep(c("Male", "Female"), each = 4), c("Male", "Female")),
  n = c(220, 200, 160, 90, 210, 205, 170, 120)
)

# Aggregated data: the value is summed
ggplot(pop, aes(x = n, y = age, fill = sex)) +
  geom_pyramid() +
  coord_pyramid()

# Individual records: rows are counted
people <- pop[rep(seq_len(nrow(pop)), pop$n), c("age", "sex")]
ggplot(people, aes(y = age, fill = sex)) +
```

```
  geom_pyramid()

# Percentages of each side
ggplot(pop, aes(x = n, y = age, fill = sex)) +
  geom_pyramid(denominator = "side") +
  scale_x_continuous(labels = scales::label_percent()) +
  coord_pyramid() +
  labs(x = "Share of each sex")
```

`geom_pyramid_outline` *A stepped outline of a population pyramid*

Description

`geom_pyramid_outline()` traces the outer edge of a pyramid as a stepped line, without filling the bars. Drawn over `geom_pyramid()`, it compares two populations, such as two census years or a region and the whole country. It aggregates the data in the same way as `geom_pyramid()`, so it accepts individual records and aggregated data alike.

Usage

```
geom_pyramid_outline(
  mapping = NULL,
  data = NULL,
  stat = "pyramid",
  position = position_pyramid(stack = FALSE),
  ...,
  denominator = NULL,
  lineend = "butt",
  linejoin = "mitre",
  linemitre = 10,
  na.rm = FALSE,
  orientation = NA,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

`mapping`	Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.

	A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
stat	The statistical transformation to use on the data for this layer. The default, "pyramid", is stat_pyramid().
position	A position adjustment. The default mirrors the left side without stacking, as position_pyramid(stack = FALSE), so that one line per side follows the total. The outline follows the side set by coord_pyramid(left = ...).
...	Other arguments passed on to ggplot2::layer(), such as fixed aesthetics like colour = "black" or linewidth = 0.8.
denominator	NULL (the default) shows counts. "total" shows each bar as a proportion of all rows in the panel, and "side" as a proportion of the rows on its side.
lineend	Line end style (round, butt, square).
linejoin	Line join style (round, mitre, bevel).
linemitre	Line mitre limit (number greater than 1).
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
orientation	The orientation of the layer. The default (NA) automatically determines the orientation from the aesthetic mapping. In the rare event that this fails it can be given explicitly by setting orientation to either "x" or "y". See the *Orientation* section for more detail.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders().

Details

Each side gets one line, and so does each subgroup that fill, colour, linetype, linewidth or alpha sets apart. The line starts at zero below the youngest age group, follows the value of each group across its whole band, and returns to zero above the oldest. It drops to zero across an age group with no data.

Value

A ggplot2 layer.

Aesthetics

The age and value axes are as in geom_pyramid(), and the side of each line comes from side, fill or colour, as described in position_pyramid(). The line is styled by colour, linewidth, linetype and alpha.

See Also

geom_pyramid() for the bars.

Examples

```
library(ggplot2)

france <- subset(europe_pop, country == "France")

# 2024 as bars, 2004 as an outline
ggplot(france, aes(x = population, y = age, fill = sex)) +
  geom_pyramid(data = subset(france, year == 2024)) +
  geom_pyramid_outline(data = subset(france, year == 2004)) +
  coord_pyramid(age_axis = "centre")
```

---

geom_pyramid_text *Value labels on a population pyramid*

Description

geom_pyramid_text() writes a label at the end of each bar of a pyramid. It aggregates the data in the same way as geom_pyramid(), so the labels show the counts, or the proportions when denominator is set, without further mapping. Labels on the left side are justified towards the left, so they sit outside (or inside) the bar on both sides.

Usage

```
geom_pyramid_text(
  mapping = NULL,
  data = NULL,
  stat = "pyramid",
  position = "pyramid",
  ...,
  denominator = NULL,
  place = c("outside", "inside"),
  padding = 1,
  labels = NULL,
  parse = FALSE,
  check_overlap = FALSE,
  size.unit = "mm",
  na.rm = FALSE,
  orientation = NA,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
stat	The statistical transformation to use on the data for this layer. The default, "pyramid", is stat_pyramid().
position	A position adjustment. The default, "pyramid", is position_pyramid(), which places each label at the outer end of its bar. Labels follow the side set by coord_pyramid(left = ...); a position given to the bars should be given to the labels as well.
...	Other arguments passed on to ggplot2::layer(), such as fixed aesthetics like size = 3 or colour = "grey30".
denominator	NULL (the default) shows counts. "total" shows each bar as a proportion of all rows in the panel, and "side" as a proportion of the rows on its side.
place	Where to draw the labels: "outside" the end of the bar, or "inside" it.
padding	The distance between the label and the end of the bar, in millimetres.
labels	A function that formats the labels, such as scales::label_comma() or scales::label_percent(), or NULL to show them as they are. A formula such as ~ round(.x) is converted with rlang::as_function().
parse	If TRUE, the labels will be parsed into expressions and displayed as described in ?plotmath.
check_overlap	If TRUE, text that overlaps previous text in the same layer will not be plotted. check_overlap happens at draw time and in the order of the data. Therefore data should be arranged by the label column before calling geom_text(). Note that this argument is not supported by geom_label().
size.unit	How the size aesthetic is interpreted: as millimetres ("mm", default), points ("pt"), centimetres ("cm"), inches ("in"), or picas ("pc").
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
orientation	The orientation of the layer. The default (NA) automatically determines the orientation from the aesthetic mapping. In the rare event that this fails it can be given explicitly by setting orientation to either "x" or "y". See the *Orientation* section for more detail.

show.legend — logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.

inherit.aes — If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders().

Value

A ggplot2 layer.

Aesthetics

label defaults to the value the bars show, after_stat(count) or after_stat(prop), and can be mapped to anything else. The horizontal and vertical justification are set by place and the text angle, so a mapped or fixed hjust or vjust is replaced. The other aesthetics are those of ggplot2::geom_text().

The justification and padding follow the orientation of the data, so ggplot2::coord_flip() puts them on the wrong axis. For a vertical pyramid, map age to x instead.

See Also

geom_pyramid() for the bars.

Examples

```
library(ggplot2)

pop <- data.frame(
  age = factor(rep(c("0-19", "20-39", "40-59", "60+"), 2)),
  sex = factor(rep(c("Male", "Female"), each = 4), c("Male", "Female")),
  n = c(220, 200, 160, 90, 210, 205, 170, 120)
)
p <- ggplot(pop, aes(x = n, y = age, fill = sex)) +
  coord_pyramid()

p +
  geom_pyramid() +
  geom_pyramid_text() +
  scale_x_continuous(expand = expansion(mult = 0.15))

# Percentages inside the bars
p +
  geom_pyramid(denominator = "total") +
  geom_pyramid_text(
    denominator = "total",
    place = "inside",
    labels = scales::label_percent(accuracy = 1),
    colour = "white"
```

```
)
```

`position_pyramid` *Mirror one side of a population pyramid*

Description

`position_pyramid()` draws the rows of one side in the negative direction, so that the two sides of a pyramid extend away from a shared age axis, and stacks subgroups within each side. It works with any layer, so bars, labels, points and lines all land on the same side.

Usage

```
position_pyramid(left = NULL, stack = TRUE, vjust = 1, reverse = FALSE)
```

Arguments

`left`	The value of the side variable drawn on the left in this layer. Overrides the `left` of `coord_pyramid()`, which sets the side for every layer at once. Without either, the first level is drawn on the left when the side variable is a factor, and the first value in sorted order otherwise.
`stack`	If `TRUE`, rows that share an age group and a side are stacked away from the centre, as in `ggplot2::position_stack()`. If `FALSE`, they overlap, which suits a comparison layer drawn over the bars. Ranges that do not start at zero, such as error bars, are never stacked.
`vjust`, `reverse`	Passed on to the stacking. See `ggplot2::position_stack()`.

Value

A ggproto object of class `PositionPyramid`.

Aesthetics

`position_pyramid()` understands the `side` aesthetic, which assigns each row to one of the two sides. When `side` is not mapped, the side comes from `fill`, and failing that from `colour`. Map `side` explicitly when `fill` shows subgroups, for example `aes(fill = region, side = sex)`. Rows with a missing side are drawn on the right.

Signed values

The data you supply stays positive. Inside the plot, values on the left side are negative, so a manual reference line or limit on that side carries a minus sign: `geom_vline(xintercept = -100)` or `scale_x_continuous(limits = c(-500, 500))`. `coord_pyramid()` shows the axis labels without the sign; with another coordinate system or in a vertical pyramid, use `scale_x_pyramid()` or `scale_y_pyramid()`.

Orientation

The axis that does not hold age is the one that gets mirrored. Age is the discrete axis when only one of them is discrete, and the y axis when both are continuous, so a pyramid is horizontal unless age is mapped to `x`. A mapped `hjust` (or `vjust` in a vertical pyramid) is mirrored as well, so text justified away from the bar end on the right does the same on the left. A fixed `hjust` set outside `ggplot2::aes()` is not mirrored.

See Also

`geom_pyramid()` for the bars of a pyramid, and `coord_pyramid()` for its axes.

Examples

```
library(ggplot2)

pop <- data.frame(
  age = factor(rep(c("0-19", "20-39", "40-59", "60+"), 2)),
  sex = factor(rep(c("Male", "Female"), each = 4), c("Male", "Female")),
  n = c(220, 200, 160, 90, 210, 205, 170, 120)
)

# Any geom can be mirrored
ggplot(pop, aes(x = n, y = age, fill = sex)) +
  geom_col(position = position_pyramid()) +
  geom_text(aes(label = n, hjust = -0.2), position = "pyramid") +
  scale_x_continuous(expand = expansion(mult = 0.15)) +
  coord_pyramid()

# Subgroups stack within each side
pop2 <- rbind(transform(pop, group = "A"), transform(pop, group = "B"))
ggplot(pop2, aes(x = n, y = age, fill = group, side = sex)) +
  geom_col(position = "pyramid") +
  coord_pyramid(left = "Female")
```

---

scale_x_pyramid *Position scales for the value axis of a pyramid*

Description

`scale_x_pyramid()` and `scale_y_pyramid()` are continuous position scales that label the value axis of a pyramid without the minus sign and extend it equally far on both sides of zero. `coord_pyramid()` does the same for a horizontal pyramid, along with placing the age axis; these scales serve a vertical pyramid, or a plot that needs another coordinate system.

Usage

```
scale_x_pyramid(..., labels = waiver(), limits = NULL, symmetric = TRUE)

scale_y_pyramid(..., labels = waiver(), limits = NULL, symmetric = TRUE)
```

Arguments

`...`	Other arguments passed on to `ggplot2::scale_x_continuous()` or `ggplot2::scale_y_continuous()` such as `breaks`, `expand` or `position`.
`labels`	Labels for the breaks, applied to their absolute values. One of: • a function that takes the breaks and returns labels, such as `scales::label_comma()`, or a formula such as `~ .x / 1000`; • the default, `waiver()`, for the usual number format; • a character vector of labels, used as given; • `NULL` for no labels.
`limits`	One of: • `NULL` to use the default scale range • A numeric vector of length two providing limits of the scale. Use `NA` to refer to the existing minimum or maximum • A function that accepts the existing (automatic) limits and returns new limits. Also accepts rlang lambda function notation. Note that setting limits on positional scales will **remove** data outside of the limits. If the purpose is to zoom, use the limit argument in the coordinate system (see `coord_cartesian()`).
`symmetric`	If `TRUE`, the limits extend equally far on both sides of zero. Ignored when `limits` is set.

Value

A ggplot2 scale.

See Also

`coord_pyramid()`, which labels and limits the value axis of a horizontal pyramid.

Examples

```
library(ggplot2)

france <- subset(europe_pop, country == "France" & year == 2024)

# A vertical pyramid, with age on x
ggplot(france, aes(x = age, y = population, fill = sex)) +
  geom_pyramid() +
  scale_y_pyramid(labels = scales::label_comma()) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```

theme_pyramid *A theme for population pyramids*

Description

theme_pyramid() is ggplot2::theme_minimal() adjusted for pyramids. It removes the grid lines that run along the bars, which the age axis already marks, keeps the major grid lines across the value axis, and puts the legend above the plot, where it reads as a heading for the two sides.

Usage

```
theme_pyramid(..., flipped = FALSE)
```

Arguments

...	Arguments passed on to ggplot2::theme_minimal(), such as base_size or base_family.
flipped	If FALSE (the default), the pyramid is horizontal, with age on the y axis. Set it to TRUE for a vertical pyramid, with age on x.

Value

A ggplot2 theme.

Examples

```
library(ggplot2)

france <- subset(europe_pop, country == "France" & year == 2024)
ggplot(france, aes(x = population, y = age, fill = sex)) +
  geom_pyramid() +
  coord_pyramid() +
  theme_pyramid()
```

Index

∗ **datasets**
    europe_pop, 4

aes(), *6*, *8*, *11*
annotation_borders(), *7*, *9*, *12*

coord_cartesian(), *15*
coord_pyramid, 2
coord_pyramid(), *7*, *13–15*

europe_pop, 4

fortify(), *6*, *8*, *11*

geom_pyramid, 5
geom_pyramid(), *8–10*, *12*, *14*
geom_pyramid_outline, 8
geom_pyramid_text, 10
ggplot(), *6*, *8*, *11*
ggplot2::aes(), *14*
ggplot2::after_stat(), *7*
ggplot2::coord_flip(), *12*
ggplot2::geom_text(), *12*
ggplot2::layer(), *6*, *9*, *11*
ggplot2::position_stack(), *13*
ggplot2::scale_x_continuous(), *3*, *15*
ggplot2::scale_y_continuous(), *15*
ggplot2::theme_minimal(), *16*

lambda, *15*

position_pyramid, 13
position_pyramid(), *2–7*, *9*, *11*

rlang::as_function(), *11*

scale_x_pyramid, 14
scale_x_pyramid(), *13*
scale_y_pyramid *(*scale_x_pyramid*)*, 14
scale_y_pyramid(), *13*
stat_pyramid *(*geom_pyramid*)*, 5
stat_pyramid(), *6*, *9*, *11*

theme_pyramid, 16