

Package ‘ggmosaic2’

September 16, 2026

Title Mosaic Plots in the 'ggplot2' Framework, Extended

Version 0.5.1

Date 2026-09-15

Description Mosaic plots in the 'ggplot2' framework. Mosaic plot functionality is provided in a single 'ggplot2' layer by calling the geom 'mosaic'. This extends the now-defunct 'ggmosaic' package by allowing fitted loglinear models which can be used to show patterns of association among variables in frequency tables.

License GPL (>= 2)

Language en-US

URL <https://friendly.github.io/ggmosaic2/>,
<https://github.com/friendly/ggmosaic2>

BugReports <https://github.com/friendly/ggmosaic2/issues>

Depends ggplot2 (>= 4.0.0), R (>= 4.1.0)

Imports productplots, dplyr, plotly (>= 4.9.3), purrr, rlang, tidyr, ggrepel, scales, withr (>= 2.5.0)

Suggests gridExtra, knitr, rmarkdown, shiny (>= 1.1.0), patchwork, vcdExtra

VignetteBuilder knitr

Encoding UTF-8

LazyData true

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Michael Friendly [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-3237-0941>),
Gavin Klorfine [aut] (ORCID: <https://orcid.org/0009-0003-0395-4512>),
Haley Jeppson [aut] (ORCID: <https://orcid.org/0000-0003-2524-4063>),
Heike Hofmann [aut] (ORCID: <https://orcid.org/0000-0001-6216-5183>),
Di Cook [aut] (ORCID: <https://orcid.org/0000-0002-3813-7155>),
Hadley Wickham [ctb] (ORCID: <https://orcid.org/0000-0003-4757-117X>)

Maintainer Michael Friendly <friendly@yorku.ca>
Repository CRAN
Date/Publication 2026-09-16 12:30:02 UTC

Contents

ddecker	2
facet_mosaic_grid	3
fly	5
GeomMosaic	7
GeomMosaicJitter	7
GeomMosaicText	7
geom_mosaic	8
geom_mosaic_jitter	13
geom_mosaic_text	17
happy	20
hbar	21
hspine	22
mosaic	22
mosaic_settings	23
product	24
scale_fill_residual	25
scale_type.productlist	27
scale_x_productlist	27
spine	30
squeeze	30
StatMosaic	31
StatMosaicJitter	31
StatMosaicText	31
theme_mosaic	32
titanic	33
vbar	33
vspine	34
Index	35

ddecker	<i>Template for a double decker plot. A double decker plot is composed of a sequence of spines in the same direction, with the final spine in the opposite direction.</i>
---------	---

Description

Template for a double decker plot. A double decker plot is composed of a sequence of spines in the same direction, with the final spine in the opposite direction.

Usage

```
ddecker(direction = "h")
```

Arguments

direction direction of first split

Value

A function of one argument, the number of splits *n*, that returns a character vector of divider function names ("hspine"/"vspine") to apply at each split – suitable for the divider argument of [geom_mosaic\(\)](#) and related layers.

Examples

```
data(titanic)
ggplot(data = titanic, aes(x = product(Class, Sex))) +
  geom_mosaic(divider = ddecker("v"))
```

facet_mosaic_grid	<i>Facet mosaic plots with panel-specific axes</i>
-------------------	--

Description

facet_mosaic_grid() lays panels out like [ggplot2::facet_grid\(\)](#), but gives every panel its own x and y position scales. Mosaic category breaks depend on the proportions calculated inside a panel, so sharing a position scale can place one panel's ticks and labels on another panel's mosaic.

Usage

```
facet_mosaic_grid(
  rows = NULL,
  cols = NULL,
  space = "fixed",
  shrink = TRUE,
  labeller = "label_value",
  as.table = TRUE,
  switch = NULL,
  drop = TRUE,
  margins = FALSE,
  axes = "all",
  axis.labels = "all"
)
```

Arguments

rows, cols	A set of variables or expressions quoted by <code>vars()</code> and defining faceting groups on the rows or columns dimension. The variables can be named (the names are passed to <code>labeller</code>). For compatibility with the classic interface, rows can also be a formula with the rows (of the tabular display) on the LHS and the columns (of the tabular display) on the RHS; the dot in the formula is used to indicate there should be no faceting on this dimension (either row or column).
space	If "fixed", the default, all panels have the same size. If "free_y" their height will be proportional to the length of the y scale; if "free_x" their width will be proportional to the length of the x scale; or if "free" both height and width will vary. This setting has no effect unless the appropriate scales also vary.
shrink	If TRUE, will shrink scales to fit output of statistics, not raw data. If FALSE, will be range of raw data before statistical summary.
labeller	A function that takes one data frame of labels and returns a list or data frame of character vectors. Each input column corresponds to one factor. Thus there will be more than one with <code>vars(cyl, am)</code>. Each output column gets displayed as one separate line in the strip label. This function should inherit from the "labeller" S3 class for compatibility with <code>labeller()</code>. You can use different labeling functions for different kind of labels, for example use <code>label_parsed()</code> for formatting facet labels. <code>label_value()</code> is used by default, check it for more details and pointers to other options.
as.table	If TRUE, the default, the facets are laid out like a table with highest values at the bottom-right. If FALSE, the facets are laid out like a plot with the highest value at the top-right.
switch	By default, the labels are displayed on the top and right of the plot. If "x", the top labels will be displayed to the bottom. If "y", the right-hand side labels will be displayed to the left. Can also be set to "both".
drop	If TRUE, the default, all factor levels not used in the data will automatically be dropped. If FALSE, all factor levels will be shown, regardless of whether or not they appear in the data.
margins	Either a logical value or a character vector. Margins are additional facets which contain all the data for each of the possible values of the faceting variables. If FALSE, no additional facets are included (the default). If TRUE, margins are included for all faceting variables. If specified as a character vector, it is the names of variables for which margins are to be created.
axes	Determines which axes will be drawn. When "margins" (default), axes will be drawn at the exterior margins. "all_x" and "all_y" will draw the respective axes at the interior panels too, whereas "all" will draw all axes at all panels.
axis.labels	Determines whether to draw labels for interior axes when the axes argument is not "margins". When "all" (default), all interior axes get labels. When "margins", only the exterior axes get labels and the interior axes get none. When "all_x" or "all_y", only draws the labels at the interior axes in the x- or y-direction respectively.

Details

Panel widths and heights remain fixed. By default, axes and their labels are drawn for every panel so that each panel's category positions are visible. The panel-specific label positions are retained with `theme_mosaic()`; that theme intentionally hides the tick marks themselves.

Value

A Facet object that can be added to a ggplot.

Author(s)

Gavin Klorfine

Examples

```
facet_data <- expand.grid(
  period = factor(c("Before", "After")),
  group = factor(c("A", "B")),
  response = factor(c("No", "Yes"))
)
facet_data$n <- c(30, 10, 20, 40, 10, 35, 40, 15)

ggplot(facet_data,
  aes(weight = n, x = product(response, group), fill = response)) +
  geom_mosaic() +
  facet_mosaic_grid(cols = vars(period))
```

fly

Flying Etiquette Survey Data

Description

Data from the results of a SurveyMonkey survey commissioned by FiveThirtyEight for the story 41 Percent of Fliers Say It's Rude To Recline Your Airplane Seat.

Usage

```
fly
```

Format

A data frame with 1040 rows and 27 variables:

id Respondent ID

flight_freq How often do you travel by plane?

do_you_recline Do you ever recline your seat when you fly?

height How tall are you?

has_child_under_18 Do you have any children under 18?

three_seats_two_arms In a row of three seats, who should get to use the two arm rests?

two_seats_one_arm In a row of two seats, who should get to use the middle arm rest?

window_shade Who should have control over the window shade?

rude_to_move_to_unsold_seat Is it rude to move to an unsold seat on a plane?

rude_to_talk_to_neighbor Generally speaking, is it rude to say more than a few words to the stranger sitting next to you on a plane?

six_hr_flight_leave_seat On a six hour flight from NYC to LA, how many times is it acceptable to get up if you're not in an aisle seat?

reclining_obligation_to_behind Under normal circumstances, does a person who reclines their seat during a flight have any obligation to the person sitting behind them?

rude_to_recline Is it rude to recline your seat on a plane?

eliminate_reclining Given the opportunity, would you eliminate the possibility of reclining seats on planes entirely?

rude_to_switch_seats_friends Is it rude to ask someone to switch seats with you in order to be closer to friends?

rude_to_switch_seats_family Is it rude to ask someone to switch seats with you in order to be closer to family?

rude_to_wake_neighbor_bathroom Is it rude to wake a passenger up if you are trying to go to the bathroom?

rude_to_wake_neighbor_walk Is it rude to wake a passenger up if you are trying to walk around?

rude_to_bring_baby In general, is it rude to bring a baby on a plane?

rude_to_bring_unruly_child In general, is it rude to knowingly bring unruly children on a plane?

use_electronics_takeoff Have you ever used personal electronics during take off or landing in violation of a flight attendant's direction?

smoked_inflight Have you ever smoked a cigarette in an airplane bathroom when it was against the rules?

gender Gender

age Age

household_income Household Income

education Education

region Region

Source

<https://github.com/fivethirtyeight/data/tree/master/flying-etiquette-survey>

GeomMosaic	<i>Geom proto</i>
------------	-------------------

Description

Geom proto

Value

A ggproto object inheriting from `ggplot2::Geom`, used internally by `geom_mosaic()` to draw the mosaic tile rectangles. Not intended to be called directly.

GeomMosaicJitter	<i>Geom proto</i>
------------------	-------------------

Description

Geom proto

Value

A ggproto object inheriting from `ggplot2::Geom`, used internally by `geom_mosaic_jitter()` to draw jittered points within mosaic plot cells. Not intended to be called directly.

GeomMosaicText	<i>Geom proto</i>
----------------	-------------------

Description

Geom proto

Value

A ggproto object inheriting from `ggplot2::Geom`, used internally by `geom_mosaic_text()` to draw text/label annotations on mosaic plot cells. Not intended to be called directly.

`geom_mosaic`*Mosaic plots.*

Description

A mosaic plot is a convenient graphical summary of the conditional distributions in a contingency table and is composed of spines in alternating directions.

Usage

```
geom_mosaic(  
  mapping = NULL,  
  data = NULL,  
  stat = "mosaic",  
  position = "identity",  
  na.rm = FALSE,  
  divider = mosaic(),  
  offset = 0.01,  
  show.legend = NA,  
  inherit.aes = TRUE,  
  expected = NULL,  
  ...  
)
```

```
stat_mosaic_text(  
  mapping = NULL,  
  data = NULL,  
  geom = "Text",  
  position = "identity",  
  na.rm = FALSE,  
  divider = mosaic(),  
  show.legend = NA,  
  inherit.aes = TRUE,  
  offset = 0.01,  
  expected = NULL,  
  ...  
)
```

```
stat_mosaic(  
  mapping = NULL,  
  data = NULL,  
  geom = "mosaic",  
  position = "identity",  
  na.rm = FALSE,  
  divider = mosaic(),  
  show.legend = NA,  
  inherit.aes = TRUE,
```

```

    offset = 0.01,
    expected = NULL,
    ...
)

```

Arguments

mapping	Set of aesthetic mappings created by aes() . If specified and <code>inherit.aes = TRUE</code> (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 <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to ggplot(). A <code>data.frame</code>, 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 <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
stat	The statistical transformation to use on the data for this layer. When using a <code>geom_*()</code> function to construct a layer, the <code>stat</code> argument can be used to override the default coupling between geoms and stats. The <code>stat</code> argument accepts the following: • A Stat ggproto subclass, for example <code>StatCount</code>. • A string naming the stat. To give the stat as a string, strip the function name of the <code>stat_</code> prefix. For example, to use <code>stat_count()</code>, give the stat as "count". • For more information and other ways to specify the stat, see the layer stat documentation.
position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The <code>position</code> argument accepts the following: • The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
na.rm	If <code>FALSE</code> (the default), removes missing values with a warning. If <code>TRUE</code> silently removes missing values.
divider	Divider function. The default divider function is <code>mosaic()</code> which will use spines in alternating directions. The four options for partitioning: • <code>vspine</code> Vertical spine partition: width constant, height varies. • <code>hspine</code> Horizontal spine partition: height constant, width varies.

	• vbar Vertical bar partition: height constant, width varies. • hbar Horizontal bar partition: width constant, height varies. When omitted, divider can be inherited from mosaic_settings().
offset	Set the fixed gap at the deepest split. Gaps increase by a factor of 1.5 toward the outermost split. When omitted, the value can be inherited from mosaic_settings() .
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() .
expected	Optional specification for loglinear model residual shading. Can be a formula (e.g., <code>~ Var1 + Var2</code>), a character shortcut ("independence", "saturated", "conditional"), or NULL (default, no model). When omitted, the value can be inherited from mosaic_settings . An explicitly supplied layer value takes priority; in particular, <code>expected = NULL</code> turns off a plot-level model for that layer. When specified, Pearson residuals are calculated and automatically mapped to fill (unless fill aesthetic is explicitly set). Use with scale_fill_residual for a diverging color scale. Positive residuals receive a solid dark blue outline and negative residuals a dashed dark red outline by default. Residuals within numerical tolerance of zero receive a solid black outline. Set <code>colour = NA</code> to remove the outlines from both the cells and the residual legend.
...	other arguments passed on to layer. These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = 'red'</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
geom	The geometric object to use to display the data for this layer. When using a <code>stat_*()</code> function to construct a layer, the <code>geom</code> argument can be used to override the default coupling between stats and geoms. The <code>geom</code> argument accepts the following: • A Geom ggproto subclass, for example <code>GeomPoint</code>. • A string naming the geom. To give the geom as a string, strip the function name of the <code>geom_</code> prefix. For example, to use <code>geom_point()</code>, give the geom as "point". • For more information and other ways to specify the geom, see the layer geom documentation.

Details

Variables mapped only to fill or alpha retain their historical role as innermost mosaic partitions, but they are not shown on the automatic product axes. Position axes label only variables explicitly mapped through `x` or `conds`. If an aesthetic variable is also included in `product()`, it remains eligible for an axis label.

Product variables are ordered from innermost to outermost. With the default mosaic divider, reversing two variables swaps their horizontal and vertical roles; for example, `product(predictions, actual)` places `actual` on the primary x axis.

Value

A ggplot2 layer that can be added to a plot.

Computed variables

x location of center of the rectangle

y location of center of the rectangle

xmin location of bottom left corner

xmax location of bottom right corner

ymin location of top left corner

ymax location of top right corner

Author(s)

Gavin Klorfine

Examples

```
data(titanic)

ggplot(data = titanic, aes(x = product(Class), fill = Survived)) +
  geom_mosaic()
# good practice: use the 'dependent' variable (or most important variable)
# as fill variable

# if there is only one variable inside `product()`,
# `product()` can be omitted
ggplot(data = titanic, aes(x = Class, fill = Survived)) +
  geom_mosaic()

ggplot(data = titanic,
       aes(x = product(Class, Age), fill = Survived)) +
  geom_mosaic()

ggplot(data = titanic,
       aes(x = product(Class), conds = product(Age), fill = Survived)) +
  geom_mosaic()

# if there is only one variable inside `product()`,
# `product()` can be omitted
ggplot(data = titanic, aes(x = Class, conds = Age, fill = Survived)) +
  geom_mosaic()

ggplot(data = titanic,
       aes(x = product(Survived, Class), fill = Age)) +
  geom_mosaic()

# Variables can be transformed directly inside mosaic aesthetics
ggplot(data = mtcars,
```

```

    aes(x = product(factor(gear)), fill = factor(cyl))) +
  geom_mosaic()

# A fill-only variable colours and partitions the tiles without appearing on
# a position axis. Reverse the product order to put `actual` on the x axis.
set.seed(19790801)
predictions <- sample(iris$Species)
confusion <- as.data.frame(table(actual = iris$Species, predictions))
confusion$is_correct <- ifelse(
  confusion$actual == confusion$predictions,
  "Correct prediction", "Incorrect prediction"
)
ggplot(confusion, aes(
  weight = Freq,
  x = product(predictions, actual),
  fill = is_correct
)) +
  geom_mosaic()

# Wrapped in donttest: still runs under R CMD check --run-donttest and is
# exercised by the package's tests to make sure it works.

data(happy)

ggplot(data = happy, aes(x = product(happy))) +
  geom_mosaic(divider = "hbar")

ggplot(data = happy, aes(x = product(happy))) +
  geom_mosaic() +
  coord_flip()

# weighting is important
ggplot(data = happy, aes(weight = wtssall, x = product(happy))) +
  geom_mosaic()

ggplot(data = happy,
  aes(weight = wtssall, x = product(health), fill = happy)) +
  geom_mosaic() +
  theme(axis.text.x=element_text(angle=35))

ggplot(data = happy,
  aes(weight = wtssall, x = product(health), fill = happy)) +
  geom_mosaic(na.rm = TRUE)

ggplot(data = happy,
  aes(weight = wtssall, x = product(health, sex, degree), fill = happy)) +
  geom_mosaic(na.rm = TRUE)

# here is where a bit more control over the spacing of the bars is helpful:
# set labels manually:
ggplot(data = happy,
  aes(weight = wtssall, x = product(age), fill = happy)) +
  geom_mosaic(na.rm = TRUE, offset = 0) +

```

```

scale_x_productlist("Age", labels=c(17+1:72))

# thin out labels manually:
labels <- c(17+1:72)
labels[labels %% 5 != 0] <- ""
ggplot(data = happy,
       aes(weight = wtssall, x = product(age), fill = happy)) +
  geom_mosaic(na.rm = TRUE, offset = 0) +
  scale_x_productlist("Age", labels=labels)

ggplot(data = happy,
       aes(weight = wtssall, x = product(age), fill = happy,
          conds = product(sex))) +
  geom_mosaic(divider = mosaic("v"), na.rm = TRUE, offset = 0.001) +
  scale_x_productlist("Age", labels=labels)

ggplot(data = happy,
       aes(weight = wtssall, x = product(age), fill = happy)) +
  geom_mosaic(na.rm = TRUE, offset = 0) +
  facet_grid(sex~.) +
  scale_x_productlist("Age", labels=labels)

ggplot(data = happy,
       aes(weight = wtssall, x = product(happy, finrela, health))) +
  geom_mosaic(divider = mosaic("h"))

ggplot(data = happy,
       aes(weight = wtssall, x = product(happy, finrela, health))) +
  geom_mosaic(offset = .005)

# Spine example
ggplot(data = happy,
       aes(weight = wtssall, x = product(health), fill = health)) +
  geom_mosaic() +
  facet_grid(happy~.)

# Residual shading with independence model
ggplot(data = titanic, aes(x = product(Class, Sex))) +
  geom_mosaic(expected = "independence") +
  scale_fill_residual()

# Custom model formula
ggplot(data = titanic, aes(x = product(Class, Sex, Survived))) +
  geom_mosaic(expected = ~ Class + Sex) +
  scale_fill_residual()
# end of donttest

```

Description

A mosaic plot with jittered dots

Usage

```
geom_mosaic_jitter(
  mapping = NULL,
  data = NULL,
  stat = "mosaic_jitter",
  position = "identity",
  na.rm = FALSE,
  divider = mosaic(),
  offset = 0.01,
  drop_level = FALSE,
  seed = NA,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)

stat_mosaic_jitter(
  mapping = NULL,
  data = NULL,
  geom = "mosaic_jitter",
  position = "identity",
  na.rm = FALSE,
  divider = mosaic(),
  show.legend = NA,
  inherit.aes = TRUE,
  offset = 0.01,
  drop_level = FALSE,
  seed = NA,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() . If specified and <code>inherit.aes = TRUE</code> (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 <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to ggplot(). A <code>data.frame</code>, 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 <code>data.frame</code>, and will be used as the layer data. A function

	can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
stat	The statistical transformation to use on the data for this layer. When using a <code>geom_*()</code> function to construct a layer, the <code>stat</code> argument can be used to override the default coupling between geoms and stats. The <code>stat</code> argument accepts the following: • A Stat ggproto subclass, for example <code>StatCount</code>. • A string naming the stat. To give the stat as a string, strip the function name of the <code>stat_</code> prefix. For example, to use <code>stat_count()</code>, give the stat as "count". • For more information and other ways to specify the stat, see the layer stat documentation.
position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The <code>position</code> argument accepts the following: • The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
na.rm	If FALSE (the default), removes missing values with a warning. If TRUE silently removes missing values.
divider	Divider function. The default divider function is <code>mosaic()</code> which will use spines in alternating directions. The four options for partitioning: • <code>vspine</code> Vertical spine partition: width constant, height varies. • <code>hspine</code> Horizontal spine partition: height constant, width varies. • <code>vbar</code> Vertical bar partition: height constant, width varies. • <code>hbar</code> Horizontal bar partition: width constant, height varies. When omitted, divider can be inherited from <code>mosaic_settings()</code>.
offset	Set the fixed gap at the deepest split. Gaps increase by a factor of 1.5 toward the outermost split. When omitted, the value can be inherited from <code>mosaic_settings()</code> .
drop_level	Generate points for the max - 1 level
seed	Random seed passed to <code>set.seed</code> . Defaults to NA, which means that <code>set.seed</code> will not be called.
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. <code>annotation_borders()</code> .

...	other arguments passed on to layer. These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = 'red'</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
geom	The geometric object to use to display the data for this layer. When using a <code>stat_*()</code> function to construct a layer, the <code>geom</code> argument can be used to override the default coupling between stats and geoms. The <code>geom</code> argument accepts the following: • A Geom ggproto subclass, for example <code>GeomPoint</code>. • A string naming the geom. To give the geom as a string, strip the function name of the <code>geom_</code> prefix. For example, to use <code>geom_point()</code>, give the geom as <code>"point"</code>. • For more information and other ways to specify the geom, see the layer geom documentation.

Details

Variables mapped only to fill, alpha, or colour remain available to the mosaic calculation and point aesthetics, but are not shown on automatic product axes. Position axes label only variables explicitly mapped through `x` or `conds`.

Value

A ggplot2 layer that can be added to a plot.

Computed variables

xmin location of bottom left corner
xmax location of bottom right corner
ymin location of top left corner
ymax location of top right corner

Author(s)

Gavin Klorfine

Examples

```
data(titanic)

ggplot(data = titanic, aes(x = product(Class))) +
  geom_mosaic(aes(fill = Survived), alpha = 0.3) +
  geom_mosaic_jitter(aes(color = Survived))

ggplot(data = titanic, aes(x = product(Class))) +
  geom_mosaic(alpha = 0.1) +
  geom_mosaic_jitter(aes(color = Survived), drop_level = TRUE)

ggplot(data = titanic, aes(x = product(Class, Sex))) +
  mosaic_settings(divider = c("vspine", "hspine", "hspine")) +
```

```

geom_mosaic(alpha = 0.3, aes(fill = Survived)) +
geom_mosaic_jitter(aes(color = Survived))

ggplot(data = titanic,
       aes(x = product(Class), conds = product(Sex), fill = Survived)) +
geom_mosaic(alpha = 0.3,
           divider = c("vspine", "hspine", "hspine")) +
geom_mosaic_jitter(
           divider = c("vspine", "hspine", "hspine"))

```

geom_mosaic_text	<i>Labeling for Mosaic plots.</i>
------------------	-----------------------------------

Description

A mosaic plot with text or labels

Usage

```

geom_mosaic_text(
  mapping = NULL,
  data = NULL,
  stat = "mosaic",
  position = "identity",
  na.rm = FALSE,
  divider = mosaic(),
  offset = 0.01,
  show.legend = NA,
  inherit.aes = TRUE,
  as.label = FALSE,
  repel = FALSE,
  repel_params = NULL,
  check_overlap = FALSE,
  display_values = "label",
  format_digits = 1,
  expected = NULL,
  ...
)

```

Arguments

mapping	Set of aesthetic mappings created by aes() . If specified and <code>inherit.aes = TRUE</code> (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 <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to ggplot() .

	A <code>data.frame</code>, 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 <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
stat	The statistical transformation to use on the data for this layer. When using a <code>geom_*()</code> function to construct a layer, the <code>stat</code> argument can be used to override the default coupling between geoms and stats. The <code>stat</code> argument accepts the following: • A <code>Stat</code> ggproto subclass, for example <code>StatCount</code>. • A string naming the stat. To give the stat as a string, strip the function name of the <code>stat_</code> prefix. For example, to use <code>stat_count()</code>, give the stat as "count". • For more information and other ways to specify the stat, see the layer stat documentation.
position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The <code>position</code> argument accepts the following: • The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
na.rm	If FALSE (the default), removes missing values with a warning. If TRUE silently removes missing values.
divider	Divider function. The default divider function is <code>mosaic()</code> which will use spines in alternating directions. The four options for partitioning: • <code>vspine</code> Vertical spine partition: width constant, height varies. • <code>hspine</code> Horizontal spine partition: height constant, width varies. • <code>vbar</code> Vertical bar partition: height constant, width varies. • <code>hbar</code> Horizontal bar partition: width constant, height varies. When omitted, divider can be inherited from mosaic_settings().
offset	Set the fixed gap at the deepest split. Gaps increase by a factor of 1.5 toward the outermost split. When omitted, the value can be inherited from mosaic_settings().
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().

<code>as.label</code>	Show as a ggplot label (box with round corners)
<code>repel</code>	Use <code>ggrepel</code> so labels don't overlap
<code>repel_params</code>	List of <code>ggrepel</code> parameters (e.g. <code>list(point.padding = 0)</code>)
<code>check_overlap</code>	If TRUE, text that overlaps previous text in the same layer will not be plotted. <code>check_overlap</code> happens at draw time and in the order of the data. Therefore data should be arranged by the label column before calling <code>geom_label()</code> or <code>geom_text()</code> .
<code>display_values</code>	Character string specifying what values to display in cells. Options: "label" (default, factor labels), "observed" (observed counts), "expected" (expected values from model), "residual" (Pearson residuals). "expected" and "residual" require a model supplied directly through <code>expected</code> or inherited from <code>mosaic_settings</code> .
<code>format_digits</code>	Number of decimal places for formatting numeric values (default: 1). Only used when <code>display_values</code> is not "label".
<code>expected</code>	Optional loglinear model specification (same as in <code>geom_mosaic</code>). An effective model is required when using <code>display_values = "expected"</code> or "residual". Supply a formula or character shortcut directly, or omit the argument to inherit a model from <code>mosaic_settings</code> . An explicitly supplied value takes priority, and NULL turns off a plot-level model for this layer.
<code>...</code>	other arguments passed on to layer. These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = 'red'</code> or <code>size = 3</code> . Text aesthetics that can be controlled include: <code>size</code> (default: 2.7), <code>colour/color</code> , <code>fontface</code> ('plain', 'bold', 'italic', 'bold.italic'), <code>family</code> (font family), <code>angle</code> (rotation in degrees), <code>hjust/vjust</code> (justification), and <code>lineheight</code> . They may also be parameters to the paired <code>geom/stat</code> .

Value

A ggplot2 layer that can be added to a plot.

Author(s)

Gavin Klorfine

Examples

```
data(titanic)

ggplot(data = titanic, aes(x = product(Class), fill = Survived)) +
  geom_mosaic() +
  geom_mosaic_text()

ggplot(data = titanic, aes(x = product(Class, Sex), fill = Survived)) +
  mosaic_settings(divider = c("vspine", "hspine", "hspine")) +
  geom_mosaic() +
  geom_mosaic_text(size = 2)

ggplot(data = happy, aes(x = product(happy, health), fill = happy)) +
  geom_mosaic(aes(x = product(health)), na.rm = TRUE, show.legend = FALSE) +
```

```

geom_mosaic_text(na.rm = TRUE, show.legend = FALSE)

# avoid overlapping text
ggplot(data = happy, aes(x = product(happy, health), fill = happy)) +
  geom_mosaic(aes(x = product(health)), na.rm = TRUE, show.legend = FALSE) +
  geom_mosaic_text(na.rm = TRUE, check_overlap = TRUE, show.legend = FALSE)

# or use ggrepel
ggplot(data = happy, aes(x = product(happy, health), fill = happy)) +
  geom_mosaic(aes(x = product(health)), na.rm = TRUE, show.legend = FALSE) +
  geom_mosaic_text(na.rm = TRUE, repel = TRUE, show.legend = FALSE)

# and as a label
ggplot(data = happy, aes(x = product(happy, health), fill = happy)) +
  geom_mosaic(aes(x = product(health)), na.rm = TRUE, show.legend = FALSE) +
  geom_mosaic_text(
    na.rm = TRUE, repel = TRUE, as.label = TRUE,
    fill = "white", show.legend = FALSE
  )

# Display observed counts in cells
ggplot(data = titanic, aes(x = product(Class, Sex))) +
  geom_mosaic(aes(fill = Survived)) +
  geom_mosaic_text(display_values = "observed")

# Display residuals with one shared model specification
ggplot(data = titanic, aes(x = product(Class, Sex))) +
  mosaic_settings(expected = "independence") +
  geom_mosaic() +
  scale_fill_residual() +
  geom_mosaic_text(display_values = "residual",
    format_digits = 2)

# Display expected values
ggplot(data = titanic, aes(x = product(Class, Sex))) +
  mosaic_settings(expected = "independence") +
  geom_mosaic() +
  scale_fill_residual() +
  geom_mosaic_text(display_values = "expected",
    format_digits = 1)

```

happy

Data related to happiness from the general social survey.

Description

The data is a small sample of variables related to happiness from the general social survey (GSS). The GSS is a yearly cross-sectional survey of Americans, run since 1972. We combine data for more than 25 years to yield over 60 thousand observations, and of the over 5,000 variables, we select some variables that are related to happiness:

Usage

```
data(happy)
```

Format

A data frame with 62466 rows and 11 variables

- year. year of the response, 1972 to 2018.
- age. age in years: 18–89 (89 stands for all 89 year olds and older).
- degree. highest education: lt high school, high school, junior college, bachelor, graduate.
- finrela. how is your financial status compared to others: far below, below average, average, above average, far above.
- happy. happiness: very happy, pretty happy, not too happy.
- health. health: excellent, good, fair, poor.
- marital. marital status: married, never married, divorced, widowed, separated.
- sex. sex: female, male.
- polviews. from extremely conservative to extremely liberal.
- partyid. party identification: strong republican, not str republican, ind near rep, independent, ind near dem, not str democrat, strong democrat, other party.
- wtssall. probability weight. 0.39–8.74

hbar

Horizontal bar partition: width constant, height varies.

Description

Horizontal bar partition: width constant, height varies.

Usage

```
hbar(data, bounds, offset = 0.02, max = NULL)
```

Arguments

data	bounds data frame
bounds	bounds of space to partition
offset	space between spines
max	maximum value

Value

A data frame of rectangle boundaries (l, r, b, t), one row per level of data.

Examples

```
data(titanic)
ggplot(data = titanic, aes(x = product(Class))) +
  geom_mosaic(divider = "hbar")
```

hspine

Horizontal spine partition: height constant, width varies.

Description

Horizontal spine partition: height constant, width varies.

Usage

```
hspine(data, bounds, offset = offset, max = NULL)
```

Arguments

data	bounds data frame
bounds	bounds of space to partition
offset	space between spines
max	maximum value

Value

A data frame of rectangle boundaries (l, r, b, t), one row per level of data.

Examples

```
data(titanic)
ggplot(data = titanic, aes(x = product(Class))) +
  geom_mosaic(divider = "hspine")
```

mosaic

Template for a mosaic plot. A mosaic plot is composed of spines in alternating directions.

Description

Template for a mosaic plot. A mosaic plot is composed of spines in alternating directions.

Usage

```
mosaic(direction = "h")
```

Arguments

direction direction of first split

Value

A function of one argument, the number of splits *n*, that returns a character vector of divider function names ("hspine"/"vspine") to apply at each split – suitable for the divider argument of [geom_mosaic\(\)](#) and related layers.

Examples

```
data(titanic)
ggplot(data = titanic, aes(x = product(Class, Sex))) +
  geom_mosaic(divider = mosaic("v"))
```

mosaic_settings	<i>Settings for mosaic plot layers</i>
-----------------	--

Description

Set the divider, gap size, and model used to calculate expected frequencies for mosaic layers in a plot. A value set directly in a layer takes priority.

Usage

```
mosaic_settings(divider, offset, expected)
```

Arguments

divider	A divider function, a character vector naming divider functions, or a list of divider functions.
offset	A single non-negative number giving the gap at the deepest split. Gaps increase by a factor of 1.5 toward the outermost split.
expected	The log-linear model used to calculate expected frequencies and Pearson residuals. Supply a formula or one of "independence", "saturated", or "conditional". The conditional model requires one or more variables mapped to conds. Supply NULL to turn off model fitting, including a model set by an earlier call to mosaic_settings() .

Details

A layer inherits a setting only when its corresponding argument is omitted. Settings are resolved in this order: an explicitly supplied layer argument (including `expected = NULL`), the plot's `mosaic_settings()` value, and the layer default (`mosaic()`, `0.01`, or `NULL`).

`divider` and `offset` are inherited by [geom_mosaic\(\)](#), [stat_mosaic\(\)](#), [geom_mosaic_text\(\)](#), [stat_mosaic_text\(\)](#), [geom_mosaic_jitter\(\)](#), and [stat_mosaic_jitter\(\)](#). `expected` is inherited by the same constructors except the mosaic-jitter geom and stat.

The position of `mosaic_settings()` among the layers in a plot has no effect. If it is added more than once, the last supplied value for each setting is used; omitted arguments do not change earlier settings. Settings are local to the plot and do not change package or session defaults. They share configuration, not computations or fitted model objects, between layers.

Value

An object of class "ggmosaic_settings" that can be added to a ggplot with `+`.

Author(s)

Gavin Klorfine

See Also

`geom_mosaic()`, `stat_mosaic()`, `geom_mosaic_text()`, `stat_mosaic_text()`, `geom_mosaic_jitter()`, `stat_mosaic_jitter()`, `mosaic()`, and `ddeck()`

Examples

```
data(titanic)

ggplot(titanic, aes(x = product(Class, Sex))) +
  mosaic_settings(
    expected = "independence",
    divider = c("vspine", "hspine"),
    offset = 0.005
  ) +
  geom_mosaic() +
  geom_mosaic_text(display_values = "residual") +
  scale_fill_residual() +
  theme_mosaic()

# An explicit layer argument overrides the plot setting. Here NULL turns
# model fitting off for the mosaic layer.
ggplot(titanic, aes(x = product(Class, Sex), fill = Survived)) +
  mosaic_settings(expected = "independence") +
  geom_mosaic(expected = NULL)
```

product

Wrapper for a list

Description

Wrapper for a list

Usage

```
product(...)
```

Arguments

... Unquoted variables going into the product plot.

Value

A list of expressions (see [rlang::exprs\(\)](#)), one per argument, used inside `aes()` to mark the variables that define the mosaic's product formula.

Examples

```
data(titanic)
ggplot(data = titanic,
       aes(x = product(Survived, Class), fill = Survived)) +
  geom_mosaic()
```

scale_fill_residual	<i>Diverging color scale for Pearson residuals</i>
---------------------	--

Description

Provides a red-white-blue color scale centered at 0 for visualizing Pearson residuals from loglinear models. Designed for use with `geom_mosaic()` when a model is supplied through the layer's expected argument or through [mosaic_settings](#).

Usage

```
scale_fill_residual(
  ...,
  low = "darkred",
  mid = "white",
  high = "darkblue",
  midpoint = 0,
  limits = NULL,
  name = "Pearson\nResidual"
)

scale_fill_residuals(
  ...,
  low = "darkred",
  mid = "white",
  high = "darkblue",
  midpoint = 0,
  limits = NULL,
  name = "Pearson\nResidual"
)
```

Arguments

<code>...</code>	Arguments passed to scale_fill_gradient2
<code>low</code>	Color for negative residuals (default: "darkred")
<code>mid</code>	Color for zero residuals (default: "white")
<code>high</code>	Color for positive residuals (default: "darkblue")
<code>midpoint</code>	Center point for color scale (default: 0)
<code>limits</code>	Range used for the color gradient. Values beyond supplied limits receive the corresponding endpoint color.
<code>name</code>	Legend title

Details

The default legend always labels -4, 0, and 4. It also labels supplied limits and the observed minimum and maximum when those differ from the limits. The legend extends to every labeled value, with solid endpoint color beyond supplied limits. When the contributing mosaic cells have outlines, positive residuals have a solid dark blue outline, negative residuals have a dashed dark red outline, and an unoutlined midpoint band (white by default) separates them at zero. Setting `colour = NA` on every contributing mosaic layer removes these outlines from both the cells and the legend. Black ticks are drawn outside the color bar, which stretches with the mosaic panel. Nearby vertical labels are separated, and a thin elbow connects each displaced label to its exact tick. The neighboring label uses a longer straight tick so nearby text shares a common alignment. Automatically generated numeric labels are rounded to one decimal place. The legend can be hidden normally with `theme(legend.position = "none")`.

Value

A ggplot2 fill scale that can be added to a plot.

Author(s)

Gavin Klorfine

Examples

```
data(titanic)

# Independence model with residual shading
ggplot(data = titanic, aes(x = product(Class, Sex))) +
  geom_mosaic(expected = "independence") +
  scale_fill_residual()

# Custom colors
ggplot(data = titanic, aes(x = product(Class, Sex))) +
  geom_mosaic(expected = "independence") +
  scale_fill_residual(low = "red", high = "blue")

# Custom limits to highlight strong deviations
ggplot(data = titanic, aes(x = product(Class, Sex, Survived))) +
```

```
geom_mosaic(expected = ~ Class + Sex) +
scale_fill_residual(limits = c(-4, 4))
```

scale_type.productlist

Helper function for determining scales

Description

Used internally to determine class of variable x

Usage

```
## S3 method for class 'productlist'
scale_type(x)
```

Arguments

x variable

Value

character string "productlist"

scale_x_productlist *Determining scales for mosaics*

Description

Determining scales for mosaics

Usage

```
scale_x_productlist(
  name = ggplot2::waiver(),
  breaks = product_breaks(),
  minor_breaks = NULL,
  labels = product_labels(),
  limits = NULL,
  expand = ggplot2::waiver(),
  oob = scales::censor,
  na.value = NA_real_,
  transform = "identity",
  position = "bottom",
  sec.axis = ggplot2::waiver()
)
```

```

scale_y_productlist(
  name = ggplot2::waiver(),
  breaks = product_breaks(),
  minor_breaks = NULL,
  labels = product_labels(),
  limits = NULL,
  expand = ggplot2::waiver(),
  oob = scales::censor,
  na.value = NA_real_,
  transform = "identity",
  position = "left",
  sec.axis = ggplot2::waiver()
)

```

ScaleContinuousProduct

Arguments

name	set to pseudo waiver function <code>product_names</code> by default.
breaks	One of: • <code>NULL</code> for no breaks • <code>waiver()</code> for the default breaks computed by the transformation object • A numeric vector of positions • A function that takes the limits as input and returns breaks as output (e.g., a function returned by <code>scales::extended_breaks()</code>). Note that for position scales, limits are provided after scale expansion. Also accepts rlang lambda function notation.
minor_breaks	One of: • <code>NULL</code> for no minor breaks • <code>waiver()</code> for the default breaks (none for discrete, one minor break between each major break for continuous) • A numeric vector of positions • A function that given the limits returns a vector of minor breaks. Also accepts rlang lambda function notation. When the function has two arguments, it will be given the limits and major break positions.
labels	One of the options below. Please note that when <code>labels</code> is a vector, it is highly recommended to also set the <code>breaks</code> argument as a vector to protect against unintended mismatches. • <code>NULL</code> for no labels • <code>waiver()</code> for the default labels computed by the transformation object • A character vector giving labels (must be same length as breaks) • An expression vector (must be the same length as breaks). See <code>?plotmath</code> for details. • A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

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()).
expand	For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.
oob	One of: • Function that handles limits outside of the scale limits (out of bounds). Also accepts rlang lambda function notation. • The default (scales::censor()) replaces out of bounds values with NA. • scales::squish() for squishing out of bounds values into range. • scales::squish_infinite() for squishing infinite values into range.
na.value	Missing values will be replaced with this value.
transform	For continuous scales, the name of a transformation object or the object itself. Built-in transformations include "asn", "atanh", "boxcox", "date", "exp", "hms", "identity", "log", "log10", "log1p", "log2", "logit", "modulus", "probability", "probit", "pseudo_log", "reciprocal", "reverse", "sqrt" and "time". A transformation object bundles together a transform, its inverse, and methods for generating breaks and labels. Transformation objects are defined in the scales package, and are called <code>transform_<name></code>. If transformations require arguments, you can call them from the scales package, e.g. scales::transform_boxcox(p = 2). You can create your own transformation with scales::new_transform().
position	For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.
sec.axis	specify a secondary axis. By default, category labels for the inner variables of the mosaic are displayed on the opposite (top or right) side whenever more than one variable is split along a direction. Set to NULL to suppress these labels, or supply a sec_axis for full control.

Value

A ggplot2 position scale that can be added to a plot.

Examples

```
data(titanic)
ggplot(data = titanic, aes(x = product(Class, Sex))) +
  geom_mosaic() +
  scale_x_productlist("Passenger class / Sex")
```

spine	<i>Spine partition: divide longest dimension.</i>
-------	---

Description

Spine partition: divide longest dimension.

Usage

```
spine(data, bounds, offset = offset, max = NULL)
```

Arguments

data	bounds data frame
bounds	bounds of space to partition
offset	space between spines
max	maximum value

Value

A data frame of rectangle boundaries (l, r, b, t), one row per level of data.

Examples

```
data(titanic)
ggplot(data = titanic, aes(x = product(Class))) +
  geom_mosaic(divider = "spine")
```

squeeze	<i>Internal helper function</i>
---------	---------------------------------

Description

Squeeze pieces to lie within specified bounds; directly copied from package productplots

Usage

```
squeeze(pieces, bounds = bound())
```

Arguments

pieces	rectangle specified via l(eft), r(ight), b(ottom), t(op)
bounds	rectangle specified via l(eft), r(ight), b(ottom), t(op)

Value

re-scaled values for piece according to boundaries given by bounds

Author(s)

Hadley Wickham

StatMosaic	<i>Geom proto</i>
------------	-------------------

Description

Geom proto

Value

A ggproto object inheriting from `ggplot2::Stat`, used internally by `stat_mosaic()` (and `geom_mosaic()`) to compute the rectangle boundaries and cell statistics for a mosaic plot layer. Not intended to be called directly.

StatMosaicJitter	<i>Geom proto</i>
------------------	-------------------

Description

Geom proto

Value

A ggproto object inheriting from `ggplot2::Stat`, used internally by `stat_mosaic_jitter()` (and `geom_mosaic_jitter()`) to compute jittered point positions for a mosaic plot layer. Not intended to be called directly.

StatMosaicText	<i>Geom proto</i>
----------------	-------------------

Description

Geom proto

Value

A ggproto object inheriting from `ggplot2::Stat`, used internally by `stat_mosaic_text()` (and `geom_mosaic_text()`) to compute label positions for a mosaic plot layer. Not intended to be called directly.

theme_mosaic	<i>Theme for mosaic plots</i>
--------------	-------------------------------

Description

Themes set the general aspect of the plot such as the color of the background, gridlines, the size and color of fonts. `theme_mosaic` provides access to the regular `ggplot2` theme, but removes any background, axes ticks, most of the gridlines, and ensures an aspect ratio of 1 for better viewing of the mosaics. This theme also applies a bold face to axes labels and allows for the convenient rotation of category labels.

Usage

```
theme_mosaic(base_size = 11, base_family = "", rot_labels = 0, ...)
```

Arguments

<code>base_size</code>	Base font size. Defaults to 11.
<code>base_family</code>	Base font family. Defaults to "", which uses the graphics device's default font family.
<code>rot_labels</code>	The angle (in degrees) used to rotate category labels. Defaults to 0 degrees, relative to the current axis orientation.
<code>...</code>	Additional arguments passed to <code>ggplot2::theme()</code> . These are applied after the mosaic theme defaults and can therefore override them.

Details

As with other `ggplot2` extensions, themes set the general look-and-feel of the plot such as the color of the background, gridlines, the size and color of fonts. `theme_mosaic()` provides access to the regular `ggplot2` theme, but: removes any background, axes ticks, most of the gridlines, and ensures an aspect ratio of 1 for better viewing of the mosaics. This theme also applies a **bold** face to axes labels and allows for the convenient rotation of category labels to avoid overlap

Value

A `ggplot2` theme object that can be added to a plot.

Author(s)

Gavin Klorfine

Examples

```
library(ggmosaic2)
data(happy)
ggplot(data = happy,
       aes(weight = wtssall, x = product(health), fill = happy)) +
  geom_mosaic(na.rm = TRUE) +
  theme_mosaic()
```

titanic

Passengers and crew on board the Titanic

Description

A dataset containing some demographics and survival of people on board the Titanic

Usage

```
titanic
```

Format

A data frame with 2201 rows and 4 variables:

Class factor variable containing the class of a passenger (1st, 2nd, 3rd) or crew.

Sex Male/Female.

Age Child/Adult. This information is not very reliable, because it was inferred from boarding documents that did not state actual age in years.

Survived Yes/No.

vbar

Vertical bar partition: height constant, width varies.

Description

Vertical bar partition: height constant, width varies.

Usage

```
vbar(data, bounds, offset = 0.02, max = NULL)
```

Arguments

data	bounds data frame
bounds	bounds of space to partition
offset	space between spines
max	maximum value

Value

A data frame of rectangle boundaries (l, r, b, t), one row per level of data.

Examples

```
data(titanic)
ggplot(data = titanic, aes(x = product(Class))) +
  geom_mosaic(divider = "vbar")
```

vspine

Vertical spine partition: width constant, height varies.

Description

Vertical spine partition: width constant, height varies.

Usage

```
vspine(data, bounds, offset = offset, max = NULL)
```

Arguments

data	bounds data frame
bounds	bounds of space to partition
offset	space between spines
max	maximum value

Value

A data frame of rectangle boundaries (l, r, b, t), one row per level of data.

Examples

```
data(titanic)
ggplot(data = titanic, aes(x = product(Class))) +
  geom_mosaic(divider = "vspine")
```

Index

- * **datasets**
 - fly, [5](#)
 - happy, [20](#)
 - titanic, [33](#)
- aes(), [9, 14, 17](#)
- annotation_borders(), [10, 15, 18](#)
- coord_cartesian(), [29](#)
- ddecker, [2](#)
- ddecker(), [24](#)
- expansion(), [29](#)
- facet_mosaic_grid, [3](#)
- fly, [5](#)
- fortify(), [9, 14, 18](#)
- geom_mosaic, [8](#)
- geom_mosaic(), [3, 7, 23, 24, 31](#)
- geom_mosaic_jitter, [13](#)
- geom_mosaic_jitter(), [7, 23, 24, 31](#)
- geom_mosaic_text, [17](#)
- geom_mosaic_text(), [7, 23, 24, 31](#)
- GeomMosaic, [7](#)
- GeomMosaicJitter, [7](#)
- GeomMosaicText, [7](#)
- ggplot(), [9, 14, 17](#)
- ggplot2::facet_grid(), [3](#)
- ggplot2::theme(), [32](#)
- happy, [20](#)
- hbar, [21](#)
- hspine, [22](#)
- label_parsed(), [4](#)
- label_value(), [4](#)
- labeller(), [4](#)
- lambda, [28, 29](#)
- layer geom, [10, 16](#)
- layer position, [9, 15, 18](#)
- layer stat, [9, 15, 18](#)
- mosaic, [22](#)
- mosaic(), [24](#)
- mosaic_settings, [10, 19, 23, 25](#)
- mosaic_settings(), [10, 15, 18](#)
- product, [24](#)
- rlang::exprs(), [25](#)
- scale_fill_gradient2, [26](#)
- scale_fill_residual, [10, 25](#)
- scale_fill_residuals
 - (scale_fill_residual), [25](#)
- scale_type.productlist, [27](#)
- scale_x_productlist, [27](#)
- scale_y_productlist
 - (scale_x_productlist), [27](#)
- ScaleContinuousProduct
 - (scale_x_productlist), [27](#)
- scales:::censor(), [29](#)
- scales:::extended_breaks(), [28](#)
- scales:::new_transform(), [29](#)
- scales:::squish(), [29](#)
- scales:::squish_infinite(), [29](#)
- sec_axis, [29](#)
- set.seed, [15](#)
- spine, [30](#)
- squeeze, [30](#)
- stat_mosaic (geom_mosaic), [8](#)
- stat_mosaic(), [23, 24, 31](#)
- stat_mosaic_jitter
 - (geom_mosaic_jitter), [13](#)
- stat_mosaic_jitter(), [23, 24, 31](#)
- stat_mosaic_text (geom_mosaic), [8](#)
- stat_mosaic_text(), [23, 24, 31](#)
- StatMosaic, [31](#)
- StatMosaicJitter, [31](#)

StatMosaicText, [31](#)

theme_mosaic, [32](#)

theme_mosaic(), [5](#)

titanic, [33](#)

transformation object, [28](#)

vars(), [4](#)

vbar, [33](#)

vspine, [34](#)