Package 'ggmosaic'

October 13, 2022

Title Mosaic Plots in the 'ggplot2' Framework

Version 0.3.3

Description Mosaic plots in the 'ggplot2' framework. Mosaic plot functionality is provided in a single 'ggplot2' layer by calling the geom 'mosaic'.

License GPL (>= 2)

URL https://github.com/haleyjeppson/ggmosaic

BugReports https://github.com/haleyjeppson/ggmosaic

Depends ggplot2 (>= 3.3.0), R (>= 3.5.0)

Imports productplots, dplyr, plotly (>= 4.5.5), purrr, rlang, tidyr, ggrepel, scales

Suggests gridExtra, knitr, NHANES, rmarkdown, patchwork

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Haley Jeppson [aut, cre],
Heike Hofmann [aut],
Di Cook [aut],
Hadley Wickham [ctb]

Maintainer Haley Jeppson <hjeppson@iastate.edu>

Repository CRAN

Date/Publication 2021-02-23 19:50:02 UTC

R topics documented:

`ddecker` *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.*

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

`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 n 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` *Geom proto*

Description

Geom proto

`GeomMosaicJitter` *Geom proto*

Description

Geom proto

`GeomMosaicText` *Geom proto*

Description

Geom proto

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 = FALSE,
  ...
)

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,
  ...
)

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

```
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or 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, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
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 mosaic() which will use spines in alternating directions. The four options for partitioning: • vspine Vertical spine partition: width constant, height varies. • hspine Horizontal spine partition: height constant, width varies. • vbar Vertical bar partition: height constant, width varies. • hbar Horizontal bar partition: width constant, height varies.
offset	Set the space between the first spine
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.
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. borders().
...	other arguments passed on to layer. These are often aesthetics, used to set an aesthetic to a fixed value, like color = 'red' or size = 3. They may also be parameters to the paired geom/stat.
geom	The geometric object to use display the data

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

Examples

```
data(titanic)

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

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

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

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

# Just excluded for timing. Examples are included in testing to make sure they work
## Not run:
data(happy)

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

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

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

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

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

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

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

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

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

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

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

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

## End(Not run) # end of don't run
```

`geom_mosaic_jitter` *Jittered dots in Mosaic plots.*

Description

A mosaic plat 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,
```

```
  show.legend = NA,
  inherit.aes = FALSE,
  ...
)

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,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or 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, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
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 mosaic() which will use spines in alternating directions. The four options for partitioning: • vspine Vertical spine partition: width constant, height varies. • hspine Horizontal spine partition: height constant, width varies. • vbar Vertical bar partition: height constant, width varies. • hbar Horizontal bar partition: width constant, height varies.
offset	Set the space between the first spine
drop_level	Generate points for the max - 1 level

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.
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. borders().
...	other arguments passed on to layer. These are often aesthetics, used to set an aesthetic to a fixed value, like color = 'red' or size = 3. They may also be parameters to the paired geom/stat.
geom	The geometric object to use display the data

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

Examples

```
data(titanic)

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

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

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

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

---

geom_mosaic_text *Labeling for Mosaic plots.*

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 = FALSE,
  as.label = FALSE,
  repel = FALSE,
  repel_params = NULL,
  check_overlap = FALSE,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or 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, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
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 mosaic() which will use spines in alternating directions. The four options for partitioning: • vspine Vertical spine partition: width constant, height varies. • hspine Horizontal spine partition: height constant, width varies. • vbar Vertical bar partition: height constant, width varies. • hbar Horizontal bar partition: width constant, height varies.
offset	Set the space between the first spine

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.
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. borders().
as.label	Show as a ggplot label (box with round corners)
repel	Use ggrepel wo labels don't overlap
repel_params	List of ggrepel parameters (e.g. list(point.padding = 0))
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_label() or geom_text().
...	other arguments passed on to layer. These are often aesthetics, used to set an aesthetic to a fixed value, like color = 'red' or size = 3. They may also be parameters to the paired geom/stat.

Examples

```
data(titanic)

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

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

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

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

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

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

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

---

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

---

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

product *Wrapper for a list*

Description

Wrapper for a list

Wrapper for a list

Usage

```
product(...)

product(...)
```

Arguments

...	Unquoted variables going into the product plot.

Examples

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

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_,
  trans = "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_,
  trans = "identity",
  position = "left",
  sec.axis = ggplot2::waiver()
)

ScaleContinuousProduct
```

Arguments

name	set to pseudo waiver function product_names by default.
breaks	One of: • NULL for no breaks • waiver() 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 `scales::extended_breaks()`)

`minor_breaks` One of:

- `NULL` for no minor breaks
- `waiver()` for the default breaks (one minor break between each major break)
- A numeric vector of positions
- A function that given the limits returns a vector of minor breaks.

`labels` One of:

- `NULL` for no labels
- `waiver()` for the default labels computed by the transformation object
- A character vector giving labels (must be same length as `breaks`)
- A function that takes the breaks as input and returns labels as output

`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 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).
- 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.

`trans` 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 `<name>_trans` (e.g., `scales::boxcox_trans()`). You can create your own transformation with `scales::trans_new()`.

`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

Format

An object of class `ScaleContinuousProduct` (inherits from `ScaleContinuousPosition`, `ScaleContinuous`, `Scale`, `ggproto`, `gg`) of length 5.

---

spine *Spine partition: divide longest dimension.*

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

---

squeeze *Internal helper function*

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 *Geom proto*

Description

Geom proto

StatMosaicJitter *Geom proto*

Description

Geom proto

StatMosaicText *Geom proto*

Description

Geom proto

theme_mosaic *Theme for mosaic plots*

Description

Themes set the general aspect of the plot such as the colour of the background, gridlines, the size and colour of fonts. `theme_mosaic` provides access to the regular ggplot2 theme, but removes any background, most of the gridlines, and ensures an aspect ratio of 1 for better viewing of the mosaics.

Arguments

`base_size`	base font size
`base_family`	base font family

Examples

```
library(ggmosaic)
data(happy)
ggplot(data = happy) +
  geom_mosaic(aes(weight=wtssall, x=product(health), fill=happy), 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

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

Index