Package ‘ggmosaic’

September 12, 2018

Title Mosaic Plots in the 'ggplot2' Framework

Version 0.2.0

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 http://github.com/haleyjeppson/ggmosaic

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

Depends ggplot2 (>= 3.0.0), R (>= 3.2.0)

Imports dplyr, plotly (>= 4.5.5), productplots (>= 0.1.1), purrr, rlang, tidyr

Suggests gridExtra, knitr, NHANES, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

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 2018-09-12 15:00:06 UTC

R topics documented:

ddecker ........................................................................................................... 2
fly .................................................................................................................... 2
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.

ddecker(direction = "h")

Arguments

direction direction of first split

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
- **FlightFreq**  How often do you travel by plane?
- **DoYouRecline**  Do you ever recline your seat when you fly?
- **Height**  How tall are you?
- **Child18**  Do you have any children under 18?
- **Seats3_2Arms**  In a row of three seats, who should get to use the two arm rests?
- **Seats2_1Arm**  In a row of two seats, who should get to use the middle arm rest?
- **WhoControlsWindowShade**  Who should have control over the window shade?
- **RudeToMoveToUnsoldSeat**  Is it rude to move to an unsold seat on a plane?
- **RudeToTalkToNeighbor**  Generally speaking, is it rude to say more than a few words to the stranger sitting next to you on a plane?
- **SixHrFlightRudeToLeaveSeat**  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?
- **RecliningObligationToBehind**  Under normal circumstances, does a person who reclines their seat during a flight have any obligation to the person sitting behind them?
- **RudeToRecline**  Is it rude to recline your seat on a plane?
- **EliminateReclining**  Given the opportunity, would you eliminate the possibility of reclining seats on planes entirely?
- **RudeToSwitchSeatsForFriends**  Is it rude to ask someone to switch seats with you in order to be closer to friends?
- **RudeToSwitchSeatsForFamily**  Is it rude to ask someone to switch seats with you in order to be closer to family?
- **RudeToWakeNeighborForBathroom**  Is it rude to wake a passenger up if you are trying to go to the bathroom?
- **RudeToWakeNeighborForWalk**  Is it rude to wake a passenger up if you are trying to walk around?
- **RudeToBringBaby**  In general, is it rude to bring a baby on a plane?
- **RudeToBringUnrulyChild**  In general, is it rude to knowingly bring unruly children on a plane?
- **UseElectronicsDuringTakeoff**  Have you ever used personal electronics during take off or landing in violation of a flight attendant’s direction?
- **HaveYouSmoked**  Have you ever smoked a cigarette in an airplane bathroom when it was against the rules?
- **Gender**  Gender
- **Age**  Age
- **HouseholdIncome**  Household Income
- **Education**  Education
- **Region**  Region

Source

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

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

```r
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.

- **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.

gem 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)
titanic <- as.data.frame(Titanic)
titanic$Survived <- factor(titanic$Survived, levels=c("Yes", "No"))

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

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

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

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

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

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)) +
```
happy

Data related to happiness from the general social survey.

Description

The General Social Survey (GSS) is a yearly cross-sectional survey of Americans, run since 1972. This data set is a small subset of the over 5000 variables collected in the GSS. We combine data since 1972 to yield more than 50 thousand observations, for some variables that are related to happiness:
Usage

data(happy)

Format

A data frame with 62466 rows and 11 variables

Details

- 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.
- wtsall. probability weight. 0.39–8.74

Source

http://gss.norc.org/Get-The-Data

---

**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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>bounds data frame</td>
</tr>
<tr>
<td>bounds</td>
<td>bounds of space to partition</td>
</tr>
<tr>
<td>offset</td>
<td>space between spines</td>
</tr>
<tr>
<td>max</td>
<td>maximum value</td>
</tr>
</tbody>
</table>
**mosaic**

---

**hspine**

*Horizontal spine partition: height constant, width varies.*

---

**Description**

Horizontal spine partition: height constant, width varies.

**Usage**

```r
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**

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

**Arguments**

- **direction** direction of first split
### product

**Description**

Wrapper for a list

**Usage**

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

**Arguments**

```r
...  
```

Unquoted variables going into the product plot.

**Examples**

```r
data(Titanic)  
titanic <- data.frame(Titanic)  
titanic$Survived <- factor(titanic$Survived, levels=c("Yes", "No"))  
ggplot(data=titanic) +  
  geom_mosaic(aes(weight=Freq, x=product(Survived, Class), fill=Survived))  
```

---

### scale_type.product

**Description**

Helper function that ggplot2 needs for determining scales on x and y

**Usage**

```r
## S3 method for class 'product'  
scale_type(x)  
```

**Arguments**

```r
x  
```

variable under consideration

**Value**

character string "product"
Helper function for determining scales

Description

Used internally to determine class of variable x

Usage

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

Arguments

- `x` variable

Value

character string "productlist"

---

Helper function for determining scales

Description

Used internally, might not needed to be exported. HH: XXXX let's check

Usage

```r
## S3 method for class 'tbl_df'
scale_type(x)
```

Arguments

- `x` variable

Value

character string "productlist"
### scale_x_productlist

**Determining scales for mosaics**

**Description**

Determining scales for mosaics

**Usage**

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

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

ScaleContinuousProduct

**Arguments**

- **name**
  The name of the scale. Used as axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

- **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

- **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**
  A numeric vector of length two providing limits of the scale. Use `NA` to refer to the existing minimum or maximum.
spine

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

**Description**

Spine partition: divide longest dimension.

**Usage**

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

**Format**

An object of class `ScaleContinuousProduct` (inherits from `ScaleContinuousPosition`, `ScaleContinuous`, `Scale`, `ggproto`, `gg`) of length 4.

**Spine partition: divide longest dimension.**

**Expand**

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 `expand_scale()` 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**

Function that handles limits outside of the scale limits (out of bounds). The default replaces out of bounds values with NA.

**NaN value**

Missing values will be replaced with this value.

**Trans**

Either the name of a transformation object, or the object itself. Built-in transformations include "asn", "atanh", "boxcox", "exp", "identity", "log", "log10", "log1p", "log2", "logit", "probability", "probit", "reciprocal", "reverse" and "sqrt". 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**

The position of the axis. "left" or "right" for vertical scales, "top" or "bottom" for horizontal scales.

**Sec.axis**

Specify a secondary axis.
**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(left), r(right), b(ottom), t(op)
- **bounds**: rectangle specified via l(left), r(right), 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

---

**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

*Topic datasets
  fly, 2
  happy, 6
  scale_x_productlist, 11
  StatMosaic, 13

aes(), 4
aes_(), 4

borders(), 5
ddecker, 2

expand_scale(), 12

fly, 2
fortify(), 4

geom_mosaic, 4
ggplot(), 4

happy, 6
hbar, 7
hspine, 8

mosaic, 8

product, 9

scale_type.product, 9
scale_type.productlist, 10
scale_type.tbl_df, 10
scale_x_productlist, 11
scale_y_productlist
  (scale_x_productlist), 11
ScaleContinuousProduct
  (scale_x_productlist), 11
scales::boxcox_trans(), 12
scales::trans_new(), 12
spine, 12
squeeze, 13

stat_mosaic (geom_mosaic), 4
StatMosaic, 13
vbar, 13
vspine, 14