chernoffGrob  

**Draw a smiley face**

**Description**

Uses Grid graphics to draw a face.

**Usage**

```r
chernoffGrob(x = 0.5, y = 0.5, size = 1, colour = "black", fill = NA,
alpha = 1, smile = 1, brow = NA, nose = FALSE)
```

**Arguments**

- `x`: horizontal position
- `y`: vertical position
- `size`: area of the face
- `colour`: colour of outlines and features
- `fill`: fill colour
- `alpha`: transparency, where 0 is transparent and 1 is opaque
- `smile`: amount of smiling/frowning
- `brow`: eyebrow angle, to represent anger or concern
- `nose`: logical. Adds a nose to the face

**Value**

A grobTree object.

**See Also**

geom_chernoff

**Examples**

```r
face <- chernoffGrob(.5, .5, size = 1e3, smile = -1, brow = 1, colour = 'navy', fill = 'lightblue')
grid::grid.newpage()
grid::grid.draw(face)
```
**geom_chernoff**

Chernoff faces in ggplot2

---

**Description**

The Chernoff geom is used to create data visualisations in the shape of human-like faces. By mapping to the relevant aesthetics, faces can appear to vary in happiness, anger, size, colour and so on.

**Usage**

```r
geom_chernoff(mapping = NULL, data = NULL, stat = "identity",
              position = "identity", na.rm = FALSE, show.legend = NA,
              inherit.aes = TRUE, ...)
```

**Arguments**

- `mapping`  
  Set of aesthetic mappings created by `aes` or `aes_`. If specified and `inherit.aes = TRUE` (the default), is combined with the default mapping at the top level of the plot. You only need to supply `mapping` if there isn’t a mapping defined for the plot.

- `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, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

- `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.

- `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.
Aesthetics

gem_chernoff understands the following aesthetics (required aesthetics are in bold):

• x
• y
• colour
• fill
• size

The following aesthetics are unique to geom_chernoff:

• smile
• brow
• nose

For details, see chernoffGrob.

References


See Also

chernoffGrob

Examples

```r
library(ggplot2)
ggplot(iris, aes(Sepal.Width, Sepal.Length, smile = Petal.Length, fill = Species)) + geom_chernoff()
ggplot(data.frame(x = 1:4,
                 y = c(3:1, 2.5),
                 z = factor(1:4),
                 w = rnorm(4),
                 n = c(rep(FALSE, 3), TRUE)
                 )) +
  aes(x, y, fill = z, size = x, nose = n, smile = w) + geom_chernoff()
```
scale_brow_continuous  Scales for angry eyebrows

Description

scale_brow lets you customise how eyebrows are generated from your data. It also lets you tweak the appearance of legends and so on. By default, brow is set to NA, in which case no eyebrows will appear (see Examples).

Usage

scale_brow_continuous(..., range = c(-1, 1), midpoint = mean)

scale_brow(..., range = c(-1, 1), midpoint = mean)

Arguments

... Other arguments passed onto continuous_scale to control name, limits, breaks, labels and so forth.
range Output range of eyebrow angles. +1 corresponds to very angry and -1 corresponds to a worried look.
midpoint A value or function of your data that will return level eyebrows, i.e. :-(

Details

Use range to vary how angrily your maximum/minimum values are represented. Minima smaller than -1 and maxima greater than +1 are possible but might look odd! You can use midpoint to set a specific 'zero' value in your data or to have eyebrow angles represented as relative to average.

The function scale_brow is an alias of scale_brow_continuous. At some point we might also want to design a scale_brow_discrete, scale_brow_manual and so on.

Legends are a work in progress. In particular, size mappings might produce odd results.

See Also

gem_chernoff, scale_smile

Examples

library(ggplot2)
p <- ggplot(iris) +
aes(Sepal.Width, Sepal.Length, fill = Species, brow = Sepal.Length) +
geom_chernoff()
p
p + scale_brow_continuous(midpoint = min)
p + scale_brow_continuous(range = c(-.5, 2))

# Only show eyebrows if 'sad', otherwise hide them
scale_smile_continuous

Scales for smiling and frowning

Description

scale_smile lets you customise how smiles are generated from your data. It also lets you tweak the appearance of legends and so on.

Usage

scale_smile_continuous(..., range = c(-1, 1), midpoint = mean)

scale_smile(..., range = c(-1, 1), midpoint = mean)

Arguments

... Other arguments passed onto continuous_scale to control name, limits, breaks, labels and so forth.

range Output range of smiles. +1 corresponds to a full smile and -1 corresponds to a full frown.

midpoint A value or function of your data that will return a neutral/straight face, i.e. :-) |

Details

Use range to vary how happily/sadly your maximum/minimum values are represented. Minima smaller than -1 and maxima greater than +1 are possible but might look odd! You can use midpoint to set a specific ‘zero’ value in your data or to have smiles represented as relative to average.

The function scale_smile is an alias of scale_smile_continuous. At some point we might also want to design a scale_smile_discrete, scale_smile_manual and so on.

Legends are a work in progress. In particular, size mappings might produce odd results.

See Also

gem_chernoff, scale_brow
Examples

```r
library(ggplot2)
p <- ggplot(iris) +
aes(Sepal.Width, Sepal.Length, fill = Species, smile = Sepal.Length) +
geom_chernoff()
p
p + scale_smile_continuous(midpoint = min)
p + scale_smile_continuous(range = c(-.5, 2))
```
Index

aes, 3
aes_, 3

borders, 3

chernoffGrob, 2, 4
continuous_scale, 5, 6

fortify, 3

ggplot, 3

geom_chernoff, 2, 3, 5, 6

Grid, 2
grobTree, 2

layer, 3

scale_brow, 6
scale_brow (scale_brow_continuous), 5
scale_brow_continuous, 5
scale_smile, 5
scale_smile (scale_smile_continuous), 6
scale_smile_continuous, 6