Package 'eulerr'

February 5, 2018

Title Area-Proportional Euler and Venn Diagrams with Circles or Ellipses

Version 4.0.0

Description Generate area-proportional Euler diagrams using numerical optimization. An Euler diagram is a generalization of a Venn diagram, relaxing the criterion that all interactions need to be represented. Diagrams may be fit with ellipses and circles via a wide range of inputs and can be visualized in numerous ways.

Depends R (>= 3.3.0)

Imports graphics, grDevices, grid, lattice, polyclip, Rcpp, RcppDE, stats, utils,

Suggests covr, knitr, pBrackets, RConics, testthat

LinkingTo Rcpp (>= 0.12.12), RcppArmadillo (>= 0.7.600.1.0)

License GPL-3

Encoding UTF-8

LazyData true

VignetteBuilder knitr

URL https://github.com/jolars/eulerr

BugReports https://github.com/jolars/eulerr/issues

RoxygenNote 6.0.1

NeedsCompilation yes

Author Johan Larsson [aut, cre],
A. Jonathan R. Godfrey [ctb],
Tim Kelley [ctb] (original Nelder-Mead code),
David H. Eberly [ctb] (geometric algorithms),
Peter Gustafsson [ctb],
Emanuel Huber [ctb] (root solver code)

Maintainer Johan Larsson <johanlarsson@outlook.com>

Repository CRAN

Date/Publication 2018-02-05 22:30:26 UTC

R topics documented:

`euler` *Area-proportional Euler diagrams*

Description

Fit Euler diagrams (a generalization of Venn diagrams) using numerical optimization to find exact or approximate solutions to a specification of set relationships. The shape of the diagram may be a circle or an ellipse.

Usage

```
euler(combinations, ...)

## Default S3 method:
euler(combinations, input = c("disjoint", "union"),
  shape = c("circle", "ellipse"), control = list(), ...)

## S3 method for class 'data.frame'
euler(combinations, weights = NULL, by = NULL, ...)

## S3 method for class 'matrix'
euler(combinations, ...)

## S3 method for class 'table'
euler(combinations, ...)

## S3 method for class 'list'
euler(combinations, ...)
```

Arguments

`combinations`	set relationships as a named numeric vector, matrix, or data.frame (see **methods (by class)**)
`...`	arguments passed down to other methods
`input`	type of input: disjoint identities (`'disjoint'`) or unions (`'union'`).
`shape`	geometric shape used in the diagram
`control`	a list of control parameters.

- `extraopt`: should the more thorough optimizer (currently `RcppDE::DEoptim()`) kick in (provided `extraopt_threshold` is exceeded)? The default is `TRUE` for ellipses and three sets and `FALSE` otherwise.
- `extraopt_threshold`: threshold, in terms of `diagError`, for when the extra optimizer kicks in. This will almost always slow down the process considerably. A value of 0 means that the extra optimizer will kick in if there is *any* error. A value of 1 means that it will never kick in. The default is `0.001`.
- `extraopt_control`: a list of control parameters to pass to the extra optimizer, such as `itermax`. See `RcppDE::DEoptim.control()`.

`weights` a numeric vector of weights of the same length as the number of rows in `combinations`.

`by` a factor or character matrix to be used in `base::by()` to split the data.frame or matrix of set combinations

Details

If the input is a matrix or data frame and argument `by` is specified, the function returns a list of euler diagrams.

The function minimizes the *stress* statistic from **venneuler**,

$$\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} y_i^2},$$

where $\hat{y}$ are ordinary least squares estimates from the regression of the fitted areas on the original areas that are currently being explored. The stress statistic can also be used as a goodness of fit measure.

`euler()` also returns `diagError` and `regionError` from **eulerAPE**. `regionError` is computed as

$$\left| \frac{y_i}{\sum y_i} - \frac{\hat{y}_i}{\sum \hat{y}_i} \right|.$$

`diagError` is simply the maximum of regionError.

Value

A list object of class `'euler'` with the following parameters.

`coefficients` a matrix of `h` and `k` (`x` and `y`-coordinates for the centers of the shapes), semiaxes `a` and `b`, and rotation angle `phi`

`original.values` set relationships in the input

`fitted.values` set relationships in the solution

`residuals` residuals

`regionError` the difference in percentage points between each disjoint subset in the input and the respective area in the output

`diagError` the largest `regionError`

`stress` normalized residual sums of squares

Methods (by class)

- `default`: a named numeric vector, with combinations separated by an ampersand, for instance `A&B = 10`. Missing combinations are treated as being 0.
- `data.frame`: a data.frame of logicals, two-level factors (see examples).
- `matrix`: a matrix that can be converted to a data.frame of logicals (as in the description above) via `base::as.data.frame.matrix()`.
- `table`: A table with `max(dim(x)) < 3`.
- `list`: a list of vectors, each vector giving the contents of that set (with no duplicates). Vectors in the list do not need to be named.

References

Wilkinson L. Exact and Approximate Area-Proportional Circular Venn and Euler Diagrams. IEEE Transactions on Visualization and Computer Graphics (Internet). 2012 Feb (cited 2016 Apr 9);18(2):321-31. Available from: http://doi.org/10.1109/TVCG.2011.56

Micallef L, Rodgers P. eulerAPE: Drawing Area-Proportional 3-Venn Diagrams Using Ellipses. PLOS ONE (Internet). 2014 Jul (cited 2016 Dec 10);9(7):e101717. Available from: http://dx.doi.org/10.1371/journal.pone.0101717

See Also

`plot.euler()`, `print.euler()`

Examples

```
# Fit a diagram with circles
combo <- c(A = 2, B = 2, C = 2, "A&B" = 1, "A&C" = 1, "B&C" = 1)
fit1 <- euler(combo)

# Investigate the fit
fit1

# Refit using ellipses instead
fit2 <- euler(combo, shape = "ellipse")

# Investigate the fit again (which is now exact)
fit2

# Plot it
plot(fit2)

# A set with no perfect solution
euler(c("a" = 3491, "b" = 3409, "c" = 3503,
        "a&b" = 120, "a&c" = 114, "b&c" = 132,
        "a&b&c" = 50))

# Using grouping via the 'by' argument through the data.frame method
dat <- data.frame(
  A = sample(c(TRUE, FALSE), size = 100, replace = TRUE),
```

```
  B = sample(c(TRUE, TRUE, FALSE), size = 100, replace = TRUE),
  gender = sample(c("Men", "Women"), size = 100, replace = TRUE),
  nation = sample(c("Sweden", "Denmark"), size = 100, replace = TRUE)
)

euler(dat, by = list(gender, nation))


dat2 <- data.frame(A = c(TRUE, FALSE, TRUE, TRUE),
                   B = c(FALSE, TRUE, TRUE, FALSE))
euler(dat2, weights = c(3, 2, 1, 1))

# Using the matrix method
mat <- cbind(A = sample(c(TRUE, TRUE, FALSE), size = 50, replace = TRUE),
             B = sample(c(TRUE, FALSE), size = 50, replace = TRUE))
euler(mat)
# The table method
plot(euler(as.table(apply(Titanic, 2:4, sum))))
# A euler diagram from a list of sample spaces (the list method)
euler(list(A = c("a", "ab", "ac", "abc"),
           B = c("b", "ab", "bc", "abc"),
           C = c("c", "ac", "bc", "abc")))
```

`eulerr_options` *Get or set global graphical parameters for eulerr*

Description

This function provides a means to set default parameters for plots produced by `plot.euler()`. Query `eulerr_options()` (without any argument) to see all the available options and read more about them in `grid::gpar()` and `graphics::par()`.

Usage

```
eulerr_options(...)
```

Arguments

`...`	objects to update the global graphical parameters for **eulerr** with.

Details

Currently, the following items will be considered:

pointsize size in pts to be used as basis for fontsizes and some margin sizes in the resulting plot

fills a list of items `fill` and `alpha`

edges a list of items `col`, `alpha`, `lex`, `lwd`, and `lty`

labels a list of items `rot`, `col`, `alpha`, `fontsize`, `cex`, `fontfamily`, `fontface`, `lineheight`, and `font`

quantities a list of items `rot`, `col`, `alpha`, `fontsize`, `cex`, `fontfamily`, `lineheight`, and `font`

strips `col`, `alpha`, `fontsize`, `cex`, `fontfamily`, `lineheight`, and `font`

legend arguments to `grid::legendGrob()` as well as `col`, `alpha`, `fontsize`, `cex`, `fontfamily`, `lineheight`, and `font`

Value

This function gets or sets updates in the global environment that are used in `plot.euler()`.

See Also

`plot.euler()`, `grid::gpar()`, `graphics::par()`

Examples

```
eulerr_options(edges = list(col = "blue"), fontsize = 10)
```

---

plot.euler *Plot area-proportional Euler diagrams*

Description

Plot Euler diagrams fit with `euler()` using `grid::Grid()` graphics. This function sets up all the necessary plot parameters and computes the geometry of the diagram. `plot.eulergram()`, meanwhile, does the actual plotting of the diagram. Please see the **Details** section to learn about the individual settings for each argument.

Usage

```
## S3 method for class 'euler'
plot(x, fills = TRUE, edges = TRUE, legend = FALSE,
  labels = identical(legend, FALSE), quantities = FALSE, strips = NULL,
  n = 200L, ..., fill_alpha, auto.key, fontface, par.settings,
  default.prepanel, default.scales, panel)
```

Arguments

`x`	an object of class `'euler'`, generated from `euler()`
`fills`	a logical, vector, or list of graphical parameters for the fills in the diagram. Vectors are assumed to be colors for the fills. See `grid::grid.path()`.
`edges`	a logical, vector, or list of graphical parameters for the edges in the diagram. Vectors are assumed to be colors for the edges. See `grid::grid.polyline()`.
`legend`	a logical scalar or list. If a list, the item `side` can be used to set the location of the legend. See `grid::grid.legend()`.
`labels`	a logical, vector, or list. Vectors are assumed to be text for the labels. See `grid::grid.text()`.

quantities	a logical, vector, or list. Vectors are assumed to be text for the quantities' labels, which by default are the original values in the input to euler(). See grid::grid.text().
strips	a list, ignored unless the 'by' argument was used in euler()
n	number of vertices for the edges and fills
...	parameters to update fills and edges with and thereby a shortcut to set these parameters
fill_alpha	deprecated
auto.key	deprecated
fontface	deprecated
par.settings	deprecated
default.prepanel	deprecated
default.scales	deprecated
panel	deprecated

Details

Most of the arguments to this function accept either a logical, a vector, or a list where

- logical values set the attribute on or off,
- vectors are shortcuts to commonly used options (see the individual parameters), and
- lists enable fine-grained control, including graphical parameters as described in grid::gpar() and control arguments that are specific to each argument.

The various grid::gpar() values that are available for each argument are:

	fills	edges	labels	quantities	strips	legend
col		x	x	x	x	x
fill	x					
alpha	x	x	x	x	x	x
lty						
lwd						
lex						
fontsize		x	x	x	x	x
cex		x	x	x	x	x
fontfamily		x	x	x	x	x
lineheight		x	x	x	x	x
font		x	x	x	x	x

Defaults for these values, as well as other parameters of the plots, can be set globally using eulerr_options().

If the diagram has been fit using the data.frame or matrix methods and using the by argument, the plot area will be split into panels for each combination of the one to two factors.

For users who are looking to plot their diagram using another package, all the necessary parameters can be collected if the result of this function is assigned to a variable (rather than printed to screen).

Value

Provides an object of class 'eulergram' , which is a description of the diagram to be drawn. plot.eulergram() does the actual drawing of the diagram.

See Also

euler(), plot.eulergram(), grid::gpar(), grid::grid.polyline(), grid::grid.path(), grid::grid.legend(), grid::grid.text()

Examples

```
fit <- euler(c("A" = 10, "B" = 5, "A&B" = 3))

# Customize colors, remove borders, bump alpha, color labels white
plot(fit,
     fills = list(fill = c("red", "steelblue4"), alpha = 0.5),
     labels = list(col = "white", font = 4))

# Add quantities to the plot
plot(fit, quantities = TRUE)

# Add a custom legend and retain quantities
plot(fit, quantities = TRUE, legend = list(labels = c("foo", "bar")))

# Plot without fills and distinguish sets with border types instead
plot(fit, fill = "transparent", lty = 1:2)

# save plot parameters to plot using some other method
diagram_description <- plot(fit)
```

---

plot.eulergram *Print (plot) Euler diagram*

Description

This function is responsible for the actual drawing of 'eulergram' objects created through plot.euler(). print.eulergram() is an alias for plot.eulergram(), which has been provided so that plot.euler() gets called automatically.

Usage

```
## S3 method for class 'eulergram'
plot(x, newpage = TRUE, ...)

## S3 method for class 'eulergram'
print(x, ...)
```

Arguments

x	an object of class 'eulergram', usually the output of plot.euler()
newpage	if TRUE, opens a new page via grid.newpage() to draw on
...	ignored

Value

A plot is drawn on the current device using grid::Grid() graphics.

---

print.euler *Print Euler diagram fits*

Description

This function is responsible for printing fits from euler() and provides a summary of the fit. Prints a data frame of the original set relationships and the fitted values as well as diagError and stress statistics.

Usage

```
## S3 method for class 'euler'
print(x, round = 3, vsep = strrep("-", 0.75 *
  getOption("width")), ...)
```

Arguments

x	'euler' object from euler()
round	number of decimal places to round to
vsep	character string to paste in between euler objects when x is a nested euler object
...	arguments passed to base::print.data.frame()

Value

Summary statistics of the fitted Euler diagram is printed to screen or a plot is generated.

See Also

euler(), base::print.data.frame()

Index