# A Case Study in Fitting Area-Proportional Euler Diagrams with Ellipses using eulerr

Johan Larsson and Peter Gustafsson

Department of Statistics, School of Economics and Management, Lund University, Lund, Sweden
johanlarsson@outlook.com
peter.gustafsson@stat.lu.se

**Abstract.** Euler diagrams are common and user-friendly visualizations for set relationships. Most Euler diagrams use circles, but circles do not always yield accurate diagrams. A promising alternative is ellipses, which, in theory, enable accurate diagrams for a wider range of input. Elliptical diagrams, however, have not yet been implemented for more than three sets or three-set diagrams where there are disjoint or subset relationships. The aim of this paper is to present eulerr: a software package for elliptical Euler diagrams for, in theory, any number of sets. It fits Euler diagrams using numerical optimization and exact-area algorithms through a two-step procedure, first generating an initial layout using pairwise relationships and then finalizing this layout using all set relationships.



## 1 Background

The *Euler diagram*, first described by Leonard Euler in 1802 [1], is a generalization of the popular *Venn diagram*. Venn and Euler diagrams both visualize set relationships by mapping areas in the diagram to relationships in the data. They differ, however, in that Venn diagrams require all intersections to be present—even if they are empty—whilst Euler diagrams do not, which means that Euler diagrams lend themselves well to be area-proportional.

Euler diagrams may be fashioned out of any closed shape, and have been implemented for triangles [2], rectangles [2], ellipses [3], smooth curves [4], polygons [2], and circles [5, 2]. The latter are most common, and for good reason, being that they are easiest to interpret [6]. Circles, however, sometimes cannot be used to produce accurate area-proportional diagrams.

With four or more sets that all intersect, for instance, exact Euler diagrams are impossible with circles, given that we require $2^4 - 1 = 15$ intersections but with four circles can yield no more than 13 [7]. A solution to this problem is offered with ellipses, which may intersect in up to four, rather than two, points, consequently yielding the necessary 15 unique areas. Elliptical Euler diagrams were first introduced with **eulerAPE** [3], which, however, only supports three sets and prohibits empty intersections.

Fitting elliptical or circular Euler diagrams must be done numerically even in the two-set case [7] where the separation required by the circles has no closed-form

Y.Sato and Z.Shams (Eds.), SetVR 2018, pp. 84-91, 2018.

solution. Many algorithms accomplish this in two steps, first finding a coarse starting layout that is finalized in a second, more thorough, algorithm. For the initial layout, the aforementioned **eulerAPE** package [8], for instance, uses an algorithm that tries to minimize the error in the three-way intersection by arranging circles representing the sets. The **venneuler** package [5], meanwhile, uses multi-dimensional scaling (MDS). The javascript package **venn.js** [9] combines a *constrained* version of the MDS algorithm from **venneuler** with a greedy algorithm.

In the final layout, we need to compute the areas of the overlaps in the diagram. Frederickson [9] (**venn.js**) and Micallef and Rodgers [3] (**eulerAPE**) have developed exact-area algorithms for circles and ellipses respectively—although the latter, as we previously covered, restricts itself to three intersecting ellipses. The parameters of the circles or ellipses are then optimized numerically to minimize a loss measure, which vary depending on implementation.

The R-package **eulerr** was created as part of a bachelor's thesis [10] and is the first package to support Euler diagrams for, in theory, any number of ellipses, regardless of subset and disjoint intersections. In this paper, we aim to demonstrate the package through a series of well-known examples from previous literature on the subject as well as a simulation study for the three-set case.

## 2 Method

**eulerr** allows input in the form of *disjoint subsets*, *unions and identities*, *a matrix of binary or boolean indices*, a *list of sample spaces*, or *a two- or three-way table*. The Euler diagram is fit in two steps: first, an initial layout is formed with circles using only the sets' pairwise relationships. Second, this layout is fine-tuned taking all intersections into consideration.

### 2.1 Initial layout

For our initial layout, we adopt a constrained version of multi-dimensional scaling (MDS) that is used in **venn.js** [9], which in turn is a modification of an algorithm from **venneuler** [5].

We begin by placing circles representing each set uniformly at random in a square space with area $\sum_{i=1}^{n} r_i^2 \pi$, where $r_i$ is the radius of the $i^{\text{th}}$ circle. The circles are initialized so that their areas are proportional to the size of their respective sets. The algorithm then moves the circles so that the separation between each pair of circles matches their respective sets' intersection. If the two sets are disjoint, however, the algorithm is indifferent to the relative locations of those circles as long as they do not intersect. The equivalent applies to subset sets: as long as the circle representing the smaller set remains within the larger circle, their locations are free to vary. In all other cases, the loss function (1) is the residual sums of squares of the separation of circles, $d$, required to obtain

accurate pairwise overlaps and the actual distance in the layout.

$$\mathcal{L}(h,k) = \sum_{1 \le i < j \le N} \begin{cases} 0 & \text{if disjoint} \\ 0 & \text{if subset} \\ \left((h_i - h_j)^2 + (k_i - k_j)^2 - d_{ij}^2\right)^2 & \text{otherwise,} \end{cases} \tag{1}$$

where $h$ and $k$ are the x and y coordinates of the centers of the circles respectively. The analytic gradient follows naturally after some algebra.

We optimize (1) using the nonlinear optimizer `nlm()` from the R core package **stats** [11], which uses a set of quasi-Newton algorithms for unconstrained minimization [12].

### 2.2 Final layout

The initial layout (section 2.1) will sometimes turn up perfect diagrams[1], but only reliably so when the diagram is accurately determined by its pairwise intersections. In the final layout, we cover the remaining cases by taking each intersection into account. We now also extend ourselves to ellipses.

To find the overlap areas, we first locate all the points of intersection of the ellipses [13], after which we can establish the overlap areas of the ellipses. We are interested only in the points that are contained within all of these ellipses, which together form a shape consisting of a convex polygon, the sides of which are made up of straight lines between consecutive points, and a set of elliptical arcs—one for each pair of points (Fig. 1).

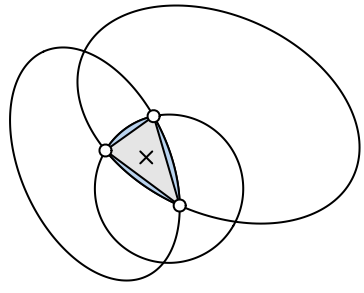

**Fig. 1.** The overlap area between three ellipses is the sum of a convex polygon (in grey) and 2–3 ellipse segments (in blue).

It is trivial to find the area of the polygon section since it is always convex [14]. And because each elliptical segment is formed from the arcs that connect successive points, it is also straightforward to establish the segments' areas [15, 16].

Having computed the areas of all the intersections we now decompose them into disjoint areas, wherein each area uniquely represents a subset of the input. This is the form we need for our final optimization. We feed the initial layout computed in section 2.1 to the optimizer—once again we employ `nlm()` from **stats** but now

[1] By perfect, we refer to solutions with diagError $< 10^{-6}$ (see equation (3)).

also provide the option to use ellipses rather than circles, allowing the "circles" to rotate and the relation between the semiaxes to vary, altogether rendering five parameters to optimize per set and ellipse (or three if we restrict ourselves to circles). For each iteration of the optimizer, the areas of all intersections are analyzed and a measure of loss returned. The loss we use is the same as that in **venneuler** [5], namely *stress*, defined as

$$\text{stress} = \frac{\sum_{i=1}^{n}(A_i - \beta\omega_i)^2}{\sum_{i=1}^{n} A_i^2} \qquad \text{with} \qquad \beta = \frac{\sum_{i=1}^{n} A_i\omega_i}{\sum_{i=1}^{n} \omega_i^2}, \tag{2}$$

where $A_i$ is the area representing $\omega_i$, the size of the $i^{\text{th}}$ *disjoint* intersection, and $n$ the number of intersections in the set configuration.

As an additional option, the user may activate a last optimization step[2] that uses a Generalized Simulated Annealing optimizer [17].

To measure the goodness of fit of the resulting diagram, we adopt two widely used measures: the previously covered *stress* [5] (2) and *diagError* [3],

$$\text{diagError} = \max_{i=1,2,\ldots,n} \left| \frac{\omega_i}{\sum_{i=1}^{n} \omega_i} - \frac{A_i}{\sum_{i=1}^{n} A_i} \right|. \tag{3}$$

The complete algorithm is provided in Algorithm 1.

### 2.3 Implementation

**eulerr** is primarily written in R but its backbone is implemented in C++ and make heavy use of the linear algebra library Armadillo [18] through **Rcpp** [19] and **RcppArmadillo** [20]. The package is compatible with all major operating systems (Linux, OS X, and Windows) and is featured on the Comprehensive R Archive Network (CRAN) [21]. It is installed by calling `install.packages("eulerr")` within R. The source code and development version are hosted at `https://github.com/jolars/eulerr`. In addition, we have developed a **shiny** [22] web application for **eulerr**, available at `http://eulerr.co`.

## 3 Results

In this section, we will study set configurations—and the diagrams fit to them—from previous papers featuring software for Euler diagrams. The packages we will study are **eulerr** 4.1.0, **eulerAPE** 3.0.0, and **venneuler** 1.1-0.

We begin with a set relationship from Wilkinson [5],

```
wilkinson <- c("A" = 4, "B" = 6, "C" = 3, "D" = 2, "E" = 7, "F" = 3,
               "A&B" = 2, "A&F" = 2, "B&C" = 2, "B&D" = 1,
               "B&F" = 2, "C&D" = 1, "D&E" = 1, "E&F" = 1,
               "A&B&F" = 1, "B&C&D" = 1)
```

[2] By default, this last-ditch optimizer kicks in for three-set combinations where the diagError of the solution surpasses 0.001.

**Algorithm 1.** The **eulerr** algorithm for elliptical diagrams.

**Data:** $N$ sets, $n$ intersections, $\omega$: the required disjoint areas for an optimal diagram, $F_i$: the size of the $i^{\text{th}}$ set, $\delta$: a predefined tolerance threshold.
**Result:** $N$ ellipses with parameters $\boldsymbol{\Psi} = \{h, k, a, b, \phi\}$.
**for** $i \leftarrow 1$ **to** $N$ **do**
  $A_i \leftarrow \omega_i$
  $r_i \leftarrow \sqrt{A_i/\pi}$
  $x_i, y_i \leftarrow \mathcal{U}\left(0, \sqrt{\sum_{i=1}^{N} r_i^2 \pi}\right)$
**foreach** $i < j \leq N$ **do**
  find $d_{ij}$ that minimizes $\left[O_{ij} - (F_i \cap F_j)\right]^2$, where $O_{ij}$ is the overlap of the circles representing $F_i$ and $F_j$
**for** $i \leftarrow 1$ **to** 10 **do** obtain $h^{(i)}$, $k^{(i)}$ by minimizing (1) using a local optimizer
$j \leftarrow i \in \{1, 10\}$ that minimizes (1)
obtain $\boldsymbol{\Psi}_{\text{final}}$ by minimizing (2) using a local optimizer with $h = h^{(j)}$, $k = k^{(j)}$, $a = 0$, $b = 0$, $\phi = 0$ as starting values
**if** `diagError(`$\boldsymbol{\Psi}_{final}$`)` $> \delta$ **then**
  obtain $\boldsymbol{\Psi}_{\text{last-ditch}}$ using a global optimizer
  **if** `diagError(`$\boldsymbol{\Psi}_{final}$`)` $>$ `diagError(`$\boldsymbol{\Psi}_{last\text{-}ditch}$`)` **then return** $\boldsymbol{\Psi}_{last\text{-}ditch}$
  **else return** $\boldsymbol{\Psi}_{final}$
**else**
  **return** $\boldsymbol{\Psi}_{final}$

specified as *disjoint subsets* using the &-operator such that `"A&B"`, for instance, are the items unique to the intersection between `A` and `B`. We fit this specification with **venneuler** and **eulerr**, in the latter case with both circles and ellipses, using `euler()`: the workhorse of **eulerr**.

```
f1 <- venneuler::venneuler(wilkinson)       # fit with venneuler
f2 <- euler(wilkinson)                      # fit with eulerr (circles)
f3 <- euler(wilkinson, shape = "ellipse")   # fit with eulerr (ellipses)
```

**eulerr** manages to fit this configuration perfectly using ellipses in addition to producing a marginally better circular diagram. The stress values are 0.007, 0.004, and $4.687 \times 10^{-12}$ for **venneuler**, **eulerr** (with circles), and **eulerr** (with ellipses) respectively.

Diagrams in **eulerr** are plotted using `plot()`, which allows considerable customization of the resulting diagram. In the following code, we plot the circular diagram using the default options and the elliptical one with a few modifications (Fig. 2).

```
plot(f2)
plot(f3,
     fills = rainbow(6, s = 0.5, v = 1),    # change fills
     edges = list(lex = 3, col = "white"), # white, broader edges
     labels = list(font = 3))              # italic labels
```

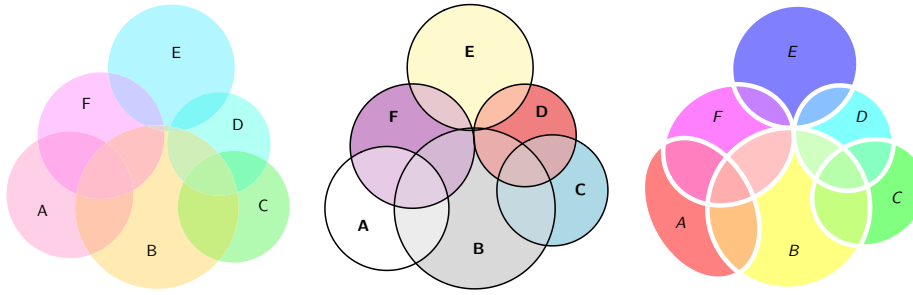


**Fig. 2.** A comparison of a Euler diagram generated with **venneuler** (to the left) with two generated from **eulerr** with circles (middle) and ellipses (right) respectively.

Micallef and Rodgers feature a diagram from Lenz et al. [23] that they remodeled using **eulerAPE** [3]. We will do the same here, using **eulerr**, and compare the results of the two packages. The data from the study—as disjoint subsets—is

```
lenz <- c("A" = 0.36, "B" = 0.03, "C" = 0,
          "A&B" = 0.41, "A&C" = 0.04, "B&C" = 0, "A&B&C" = 0.11)
```

Because **eulerAPE** cannot fit set configurations with empty intersections, the authors used 0.00001 as a proxy for $\emptyset$. Using **eulerr**, however, we can fit the diagram using the original data.

```
plot(euler(lenz, shape = "ellipse"), legend = TRUE) # add a legend
```

The fits from both packages are exact (Fig. 3). Although we instructed **eulerr** to allow ellipses in the fit, the algorithm stuck to circles, which, given that the fit is exact, is the appropriate choice since circles are easier to interpret [6]. **eulerAPE**, in contrast, did not. It tries to keep the three shapes intersecting, albeit marginally, which cannot be done with circles if the layout is to be exact.

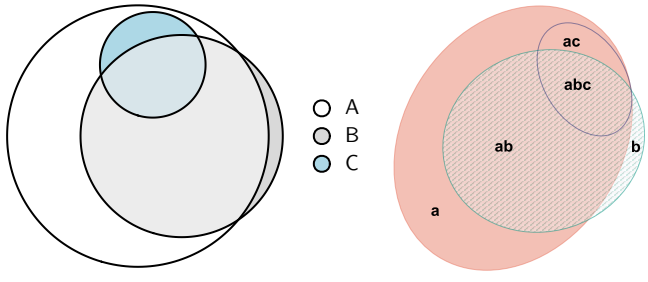


**Fig. 3.** Diagrams from **eulerAPE** and **eulerr** based on data from a diagram from Lenz et al. [23]. Both diagrams are exact.

Finally, we provide a benchmark of the accuracy of **venneuler**, **eulerAPE**, and **eulerr** in reproducing 1,000 random three-set combinations sampled from

$\mathcal{U}(10^{-6}, 1)$ (Fig. 4). We use three-set combinations with all intersections present in order to enable all tested packages to fit the diagrams.

In terms of both stress and diagError, the elliptical diagrams from **eulerr** and **eulerAPE** perform the best, with the former coming out marginally ahead with median stress and diagError at $3.123 \times 10^{-13}$ and $1.638 \times 10^{-7}$ respectively, whilst the equivalent figures for **eulerAPE** are $7.834 \times 10^{-12}$ and $8.219 \times 10^{-7}$.

For the circular diagrams, **eulerr** achieves the lowest median stress at 0.022, followed by **venneuler** and **eulerAPE** at 0.035 and 0.08 respectively. In terms of diagError, **venneuler** performs best followed by **eulerr** and **eulerAPE** with respective median diagErrors of 0.048, 0.055, and 0.067.

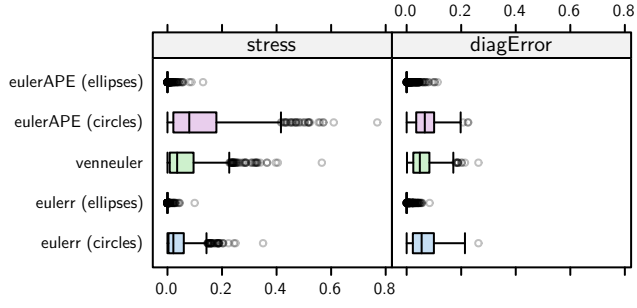


**Fig. 4.** Tukey box plots of diagError and stress for Euler diagrams based on set relationships of three sets with every intersection present.

## 4 Discussion

In this paper, we have presented an R-based software package, **eulerr**, for generating elliptical Euler diagrams for any number of sets. We have examined its performance for set relationships from previous publications of software for Euler diagrams and shown that **eulerr** performs adequately for our examples and for random three-set combinations. In general, we have also shown that elliptical Euler diagrams have the potential to outperform circular diagrams. The reason for this is simple: elliptical Euler diagrams feature two additional degrees of freedom for each shape in the diagram, provided by stretch and rotation.

**eulerr** is the first software to feature area-proportional elliptical Euler diagrams for more than three sets. The only other software for elliptical Euler diagrams, **eulerAPE**, is restricted to three sets. This limitation is discussed by the authors of the package, who argue that Euler diagrams with more than three sets often lack well-formed solutions and that their complexity make implementations difficult [8]. Whilst it is true that inputs with more than three sets do not always reduce to adequate Euler diagrams, it is our stance that those that *do* warrant a solution to find them.

The results of this paper are limited to a few cases and it is not known whether they generalize to other set combinations. This is a topic for future research in the field, which should examine different software for Euler diagrams in large-scale simulation studies.

## References


1. Euler, L.: Letters of Euler to a German princess, on different subjects in physics and philosophy. Murray and Highley (1802)
2. Swinton, J.: Vennerable: Venn and Euler area-proportional diagrams. (2011) R package version 3.1.0.9000.
3. Micallef, L., Rodgers, P.: eulerAPE: drawing area-proportional 3-Venn diagrams using ellipses. PLOS ONE **9**(7) (July 2014) e101717
4. Micallef, L., Rodgers, P.: eulerForce: force-directed layout for Euler diagrams. Journal of Visual Languages and Computing **25**(6) (December 2014) 924–934
5. Wilkinson, L.: Exact and approximate area-proportional circular Venn and Euler diagrams. IEEE Transactions on Visualization and Computer Graphics **18**(2) (February 2012) 321–331
6. Blake, A.: The impact of graphical choices on the perception of Euler diagrams. Ph.D. dissertation, Brighton University, Brighton, UK (February 2016)
7. Chow, S.C.: Generating and drawing area-proportional Euler and Venn diagrams. Ph.D. dissertation, University of Victoria, Victoria, BC, Canada (2007)
8. Micallef, L.: Visualizing set relations and cardinalities using Venn and Euler diagrams. Ph.D. dissertation, University of Kent (September 2013)
9. Frederickson, B.: venn.js: area proportional Venn and Euler diagrams in JavaScript (November 2016) original-date: 2013-05-09.
10. Larsson, J.: eulerr: Area-proportional Euler diagrams with ellipses (2018) Available at: http://lup.lub.lu.se/student-papers/record/8934042.
11. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2017)
12. Schnabel, R.B., Koonatz, J.E., Weiss, B.E.: A modular system of algorithms for unconstrained minimization. ACM Trans Math Softw **11**(4) (December 1985) 419–440
13. Richter-Gebert, J.: Perspectives on Projective Geometry: A Guided Tour Through Real and Complex Geometry. 1 edn. Springer, Berlin, Germany (February 2011)
14. Finley, D.R.: Ultra-easy algorithm with C code sample. http://alienryderflex.com/polygon_area/ (December 2006)
15. Eberly, D.: The area of intersecting ellipses (November 2016)
16. Micallef, L., Rodgers, P.: Computing the Region Areas of Euler Diagrams Drawn with Three Ellipses. In Burton, J., Stapleton, G., Klein, K., eds.: CEUR Workshop Proceedings. Volume 1244., Melbourne, Australia (July 2014) 1–15
17. Xiang, Y., Gubian, S., Suomela, B., Hoeng, J.: Generalized simulated annealing for global optimization: the GenSA package. The R Journal **5**(1) (June 2013) 13–28
18. Sanderson, C., Curtin, R.: Armadillo: a template-based C++ library for linear algebra. The Journal of Open Source Software **1**(2) (2016) 26
19. Eddelbuettel, D., François, R.: Rcpp: Seamless R and C++ integration. Journal of Statistical Software **40**(8) (2011) 1–18
20. Eddelbuettel, D., Sanderson, C.: RcppArmadillo: accelerating R with high-performance C++ linear algebra. Computational Statistics and Data Analysis **71** (March 2014) 1054–1063
21. R Core Team: The Comprehensive R Archive Network (November 2017)
22. Chang, W., Cheng, J., Allaire, J., Xie, Y., McPherson, J.: shiny: Web Application Framework for R. (2017) R package version 1.0.5.
23. Lenz, O., Fornoni, A.: Chronic kidney disease care delivered by US family medicine and internal medicine trainees: results from an online survey. BMC medicine **4** (December 2006) 30