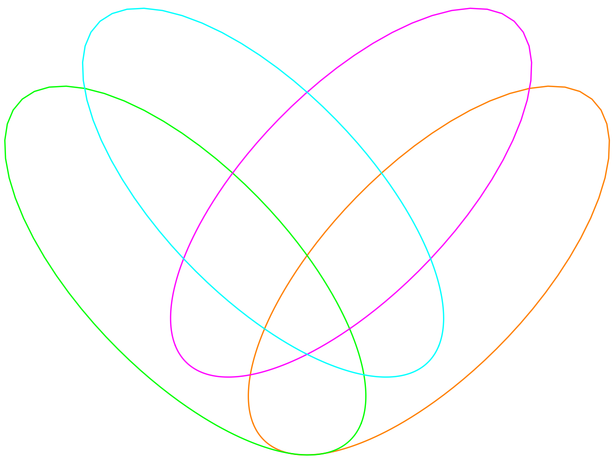
Venn diagrams with the Vennerable package

Jonathan Swinton

19th June, 2007



1 Quick tour

The Vennerable package provides routines to compute and plot Venn diagrams, from the classic two- and three-circle diagrams to diagrams using different shapes and for up to seven sets. In addition it can plot diagrams in which the area of each region is proportional to the corresponding number of set items or other weights.

Figure 1 shows a three-circle Venn diagram of the sort commonly found. First we construct an object of class `Venn`:

```
> library(Vennerable)
> Vcombo <- Venn(SetNames = c("Female",
+     "Visible Minority", "CS Major"),
+     Weight = c(0, 4148, 409, 604,
+         543, 67, 183, 146))
```

This returns an object of (S4) class `Venn` and a call to plot plot.Venn produces the diagram.

```
> plot(Vcombo, doWeights = FALSE)
```

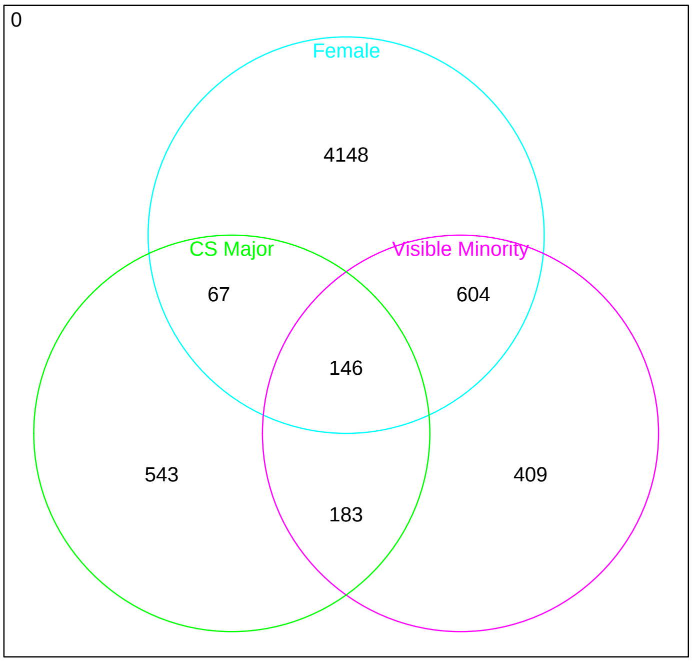


Figure 1: A three-circle Venn diagram

The `Vennerable` package extends this plot in three ways. First it allows the use of a variety of other shapes for the set boundaries, and up to seven different sets. Secondly it implements a number of published or novel algorithms for generating diagrams in which the area of each region is proportional to, for example, the number of corresponding set elements. Finally it adds a number of graphical control abilities, including the ability to colour individual regions separately.

2 Some loose definitions

Figure 1 illustrates membership of three sets, in order Female, Visible Minority, CS Major. People who are members of the Visible Minority set are not Female nor a CS Major are members of an *intersection subset* with *indicator string* `010`.

Given n sets of elements drawn from a universe, there are 2^n intersection subsets. Each of these is a subset of the universe and there is one corresponding to each of the binary strings of length n. If one of these indicator strings has a 1 in the i-th position, all of members of the corresponding intersection subset must be members of the i-th set. Depending on the application, the universe of elements from which members of the sets are drawn may be important. Elements in intersection set `00...`, which are in the universe but not in any known set, are called *dark matter*, and we tend to display these differently.

A diagram which produces a visualisation of each of the sets as a connected curve in the plane whose regions of intersection are connected and correspond to each of the 2^n intersection subsets is an *unweighted Venn diagram*. Weights can be assigned to each of the intersections, most naturally being proportional to the number of elements each one contains. *Weighted Venn diagrams* have the same topology as unweighted ones, but (attempt to) make the area of each region proportional to the weights. This may not be possible, if any of the weights are zero for example, or because of the geometric constraints of the diagram. Venn diagrams based on 3 circles are unable in general to represent even nonzero weights exactly, and cannot be constructed at all for $n > 3$.

Diagrams in which only those intersections with non-zero weight appear are *Euler diagrams*, and diagrams which go further and make the area of every intersection proportional to its weight are weighted Euler diagrams. For more details and rather more rigour see first the online review of Ruskey and Weston[5].

3 Unweighted Venn diagrams

For a running example, we use sets named after months, whose elements are the letters of their names.

```
> setList <- strsplit(month.name, split = "")
> names(setList) <- month.name
> VN3 <- VennFromSets(setList[1:3])
> V2 <- VN3[, c("January", "February"),
+     ]
```

More details on the construction of `Venn` objects is given in Section 6.

3.1 Unweighted 2-set Venn diagrams

The geometry of the diagram is controlled with the `type` argument which, for two sets, can be `type=circles` or `type=squares`. The full code for producing the Figure can be found by consulting the source of this vignette `Venn.Rnw` in the `inst/doc` subdirectory of the `Vennerable` package.

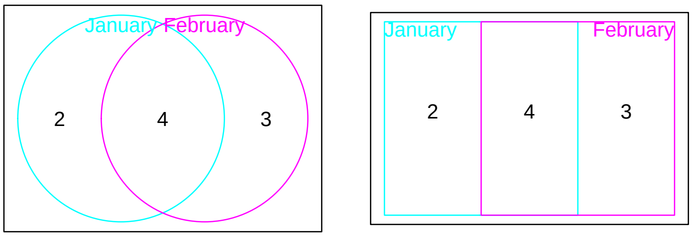
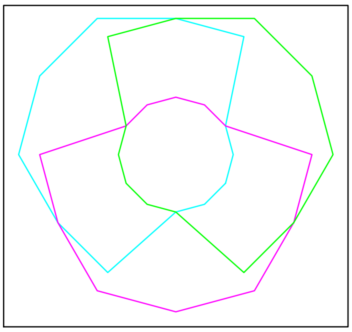


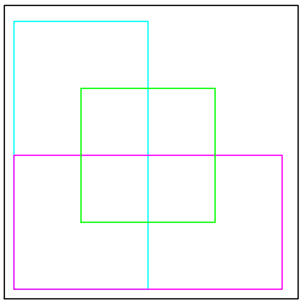
Figure 2: Unweighted 2-set Venn diagrams with `type=circles` or `type=squares`

3.2 Unweighted 3-set Venn diagrams

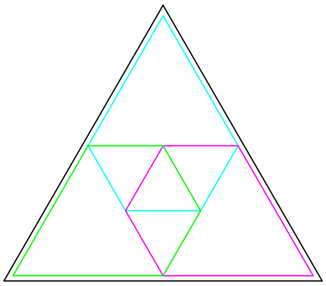
For three sets, the `type` argument can be `circles`, `squares`, `ChowRuskey`, `triangles` or `AWFE`. The Chow-Ruskey plot is from [3]. The AWFE plot is a somewhat ugly implementation of the elegant ideas of [4]. The triangles plot is new as far as I know.



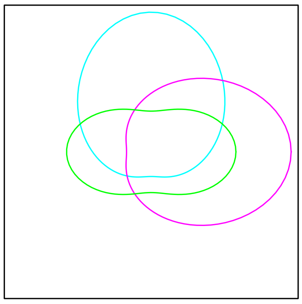
plot(VN3,type="ChowRuskey",...)



plot(VN3,type="squares",...)



plot(VN3,type="triangles",...)



plot(VN3,type="AWFE",...)

3.3 Unweighted 4-set Venn diagrams

For four sets, the `type` argument can be `ChowRuskey`, `AWFE`,`squares` or `ellipses`.

The squares plot is said by Edwards [4] to have been introduced by Lewis Carroll [1]. The ellipse plot was suggested by Venn [6].

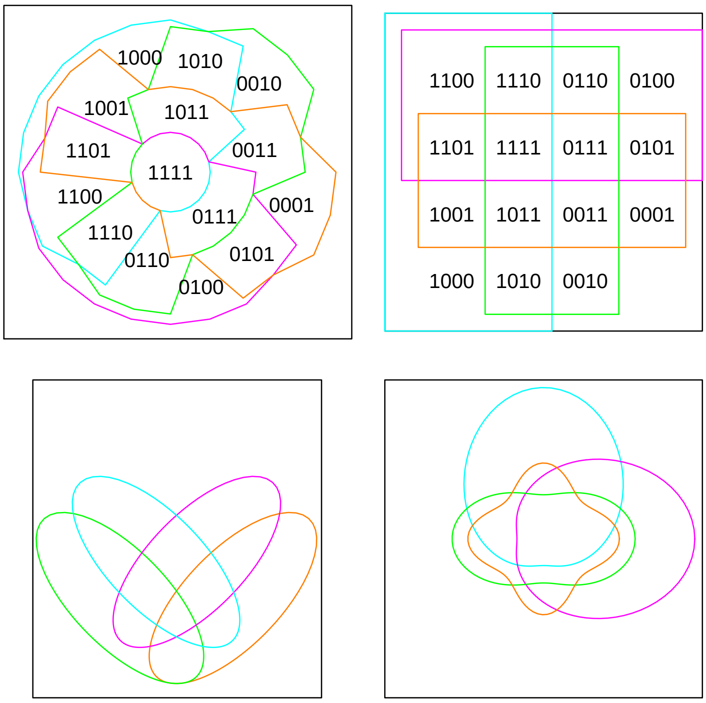


Figure 3: Venn diagrams on four sets.

A number of variants on the `squares` type are implemented, but require finer control over the diagram creation process.

```
> doans <- function(V4, s, likeSquares) {
+     S4 <- compute.S4(V4, s = s, likeSquares = likeSquares)
+     anlay <- grid.layout(2, 1, heights = unit(c(1,
+         1), c("null", "lines")))
+     pushViewport(viewport(layout = anlay))
+     txt <- ""
+     pushViewport(viewport(layout.pos.row = 2))
+     grid.text(label = txt)
+     popViewport()
+     pushViewport(viewport(layout.pos.row = 1))
+     CreateViewport(S4)
+     PlotSetBoundaries(S4, gp = gpar(lwd = 4:1,
+         col = trellis.par.get("superpose.symbol")$col))
+     UpViewports()
+     popViewport()
```

```
+      popViewport()
+ }
```

For more details on this see the help pages and 7.

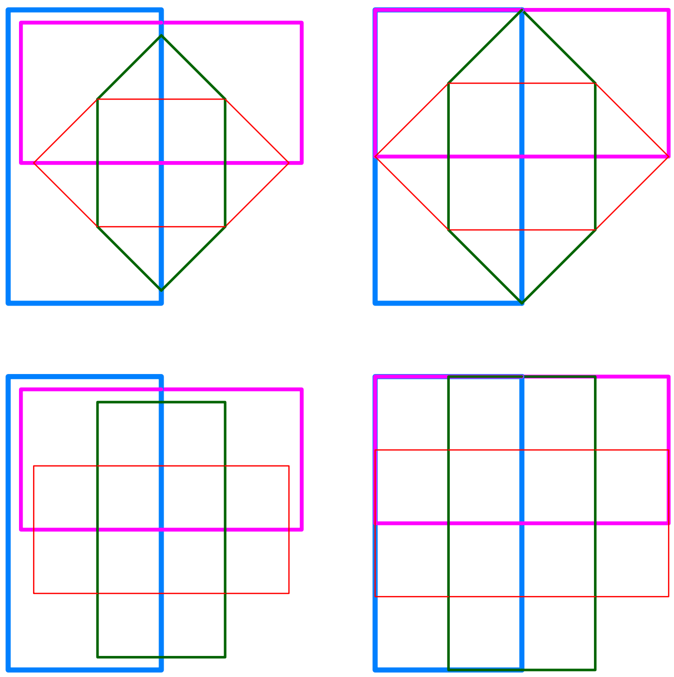


Figure 4: Four variants on the four-squares

3.4 Unweighted Venn diagrams on more than four sets

The package implements a (somewhat ugly) variant of the Edwards construction for Venn diagrams on an arbitrary number of sets n. The currently implemented algorithm is rather inefficient and is only feasible to compute quickly for $n < 7$.

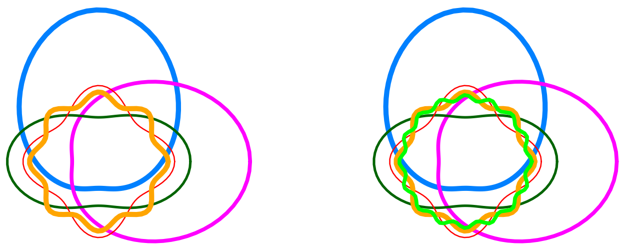


Figure 5: Edwards constructions for five and six sets

4 Weighted Venn diagrams

In this section we show Venn diagrams with weights which are (with the possible exception of dark matter) all nonzero.

4.1 Weighted 2-set Venn diagrams for 2 Sets

4.1.1 Circles

It is always possible to get an exactly area-weighted solution for two circles as shown in Figure 6.

Figure 6: Weighted 2d Venn

4.1.2 Squares

As for circles, square Venn and Euler diagrams on two sets can be simply constructed.

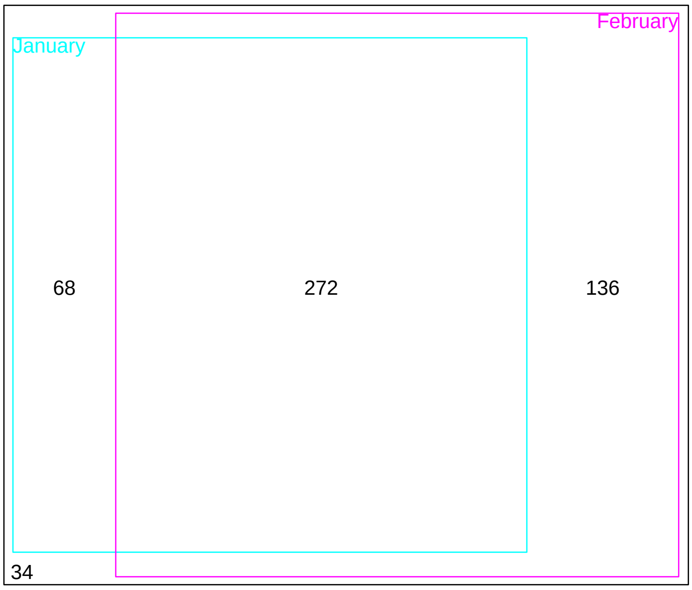


Figure 7: Weighted 2d Venn squares

4.2 Weighted 3-set Venn diagrams

4.2.1 Circles

There is no general way of creating area-proportional 3-circle diagrams. The package makes an attempt to produce approximate ones.

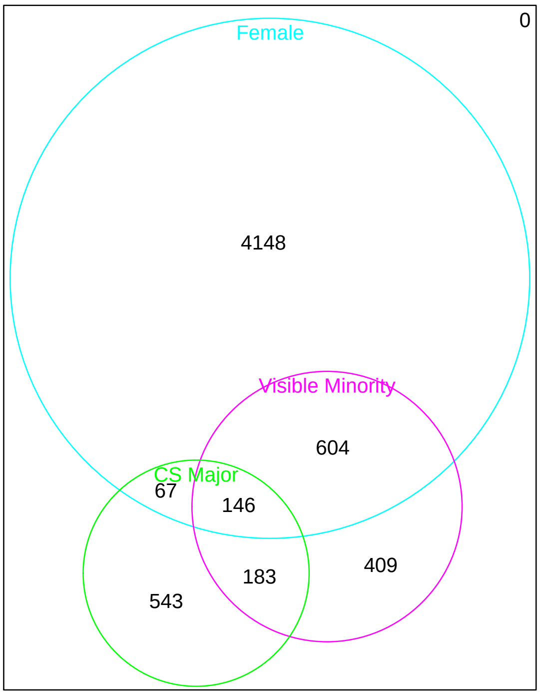


Figure 8: 3D Venn diagram

The algorithm used is to compute the individual circles to have the exact area necessary for proportionality, and then compute each of the three pairwise distances between centres necessary for the correct pairwise areas. If these distances do not satisfy the triangle inequality the largest is reduced until they do. Then the circles are arranged with their centres separated by these (possibly modified) distances.

4.2.2 Squares

This is a version of the algorithm suggested by [2].

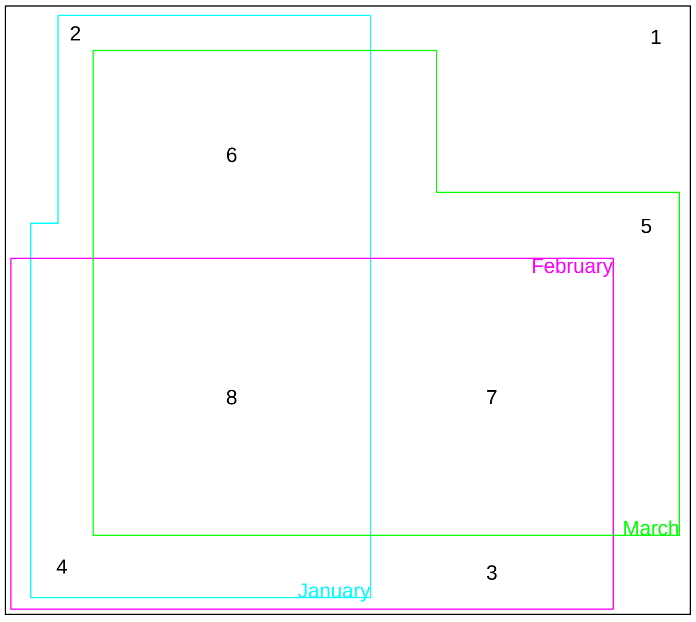


Figure 9: Weighted 3-set Venn diagram based on the algorithm of [2]

4.2.3 Triangles

The triangular Venn diagram on 3-sets lends itself nicely to an area-proportional drawing under some contraints on the weights. These constraints are that

$$4w_a w_b w_c < (1 - (w_a + w_b + w_c))^2 \tag{1}$$

must hold for both of the sets of numbers

$$\begin{aligned} w_a &= w_{100} && (2)\\ w_b &= w_{010} && (3)\\ w_c &= w_{001} && (4) \end{aligned}$$

and

$$\begin{aligned} w_a &= w_{101}/W && (5)\\ w_b &= w_{011}/W && (6)\\ w_c &= w_{011}/W && (7) \end{aligned}$$

where w_s is the normalised weight of the set with indicator string s and $W = w_{101} + w_{011} + w_{011} + w_{111} = 1 - (w_{100} + w_{010} + w_{001})$.

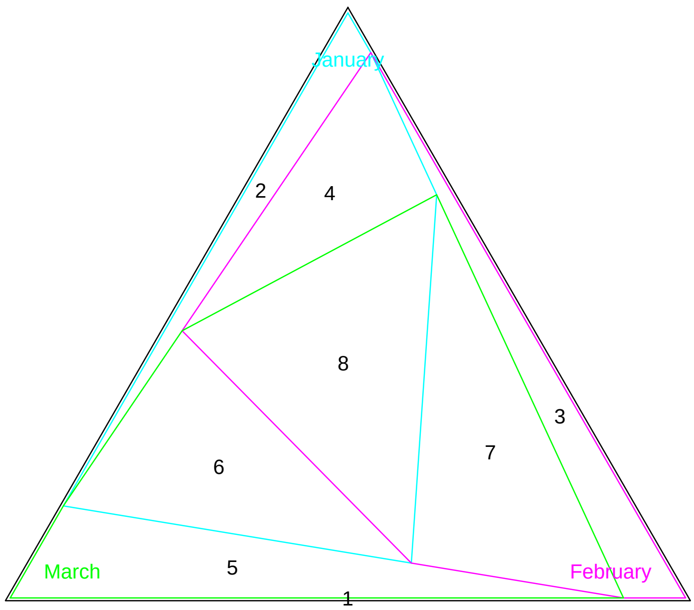


Figure 10: Triangular Venn with external universe

4.3 Chow-Ruskey diagrams for 3 or more sets

The general Chow-Ruskey algorithm [3] can be implemented in principle for an arbitrary number of sets provided the weight of the common intersection is nonzero.

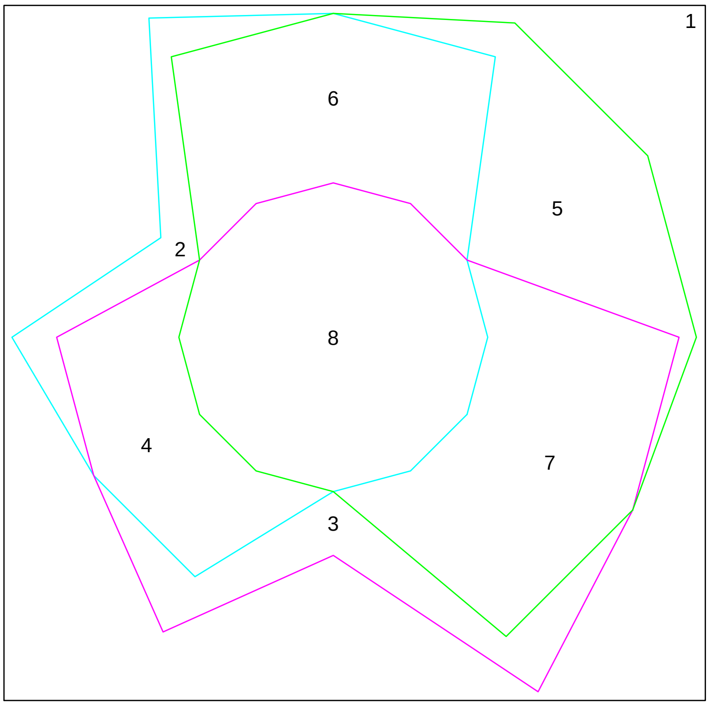


Figure 11: Chow-Ruskey weighted 3-set diagram

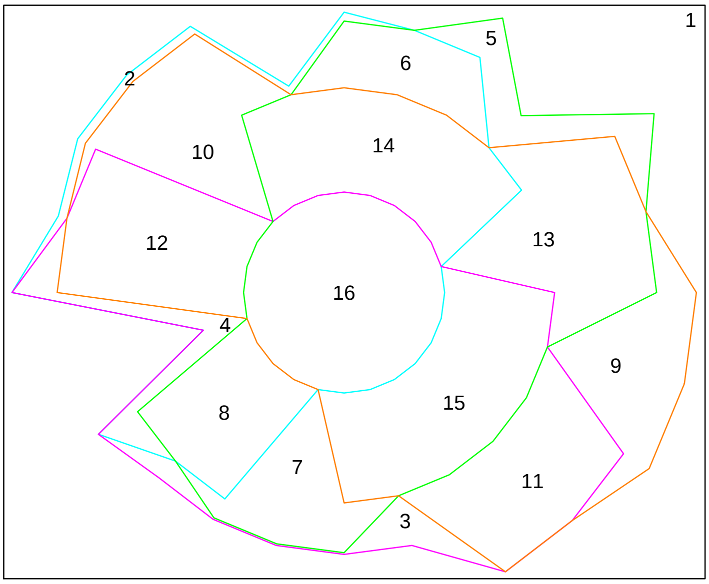


Figure 12: Chow-Ruskey weighted 4-set diagram

5 Euler diagrams

As we have seen, for some geometries it is not possible to enforce exact area-proportionality when requested by the `doWeight=TRUE` argument, and an attempt is made to produce an approximately area-proportional diagram. In particular, regions whose weight is zero may appear with nonzero areas. A separate requirement which is desirable in some cases, especially high dimensional ones, is to ensure that regions of zero weight are not displayed at all, producing an *Euler diagram*. This can be achieved, for some geometries, by use of the `doEuler=TRUE` argument. These two flags can interact in weird and uncomfortable ways depending on exactly which intersection weights are zero.

5.1 2-set Euler diagrams

5.1.1 Circles

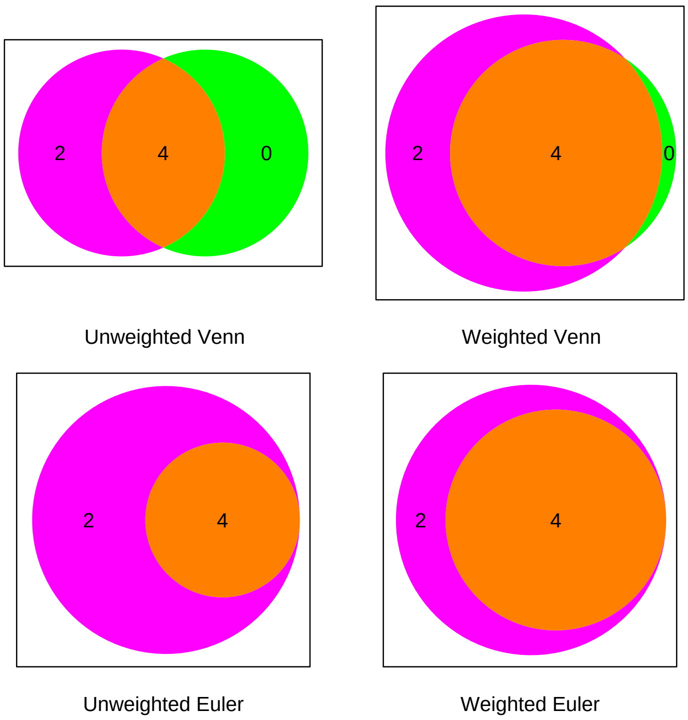


Figure 13: Effect of the Euler and `doWeights` flags.

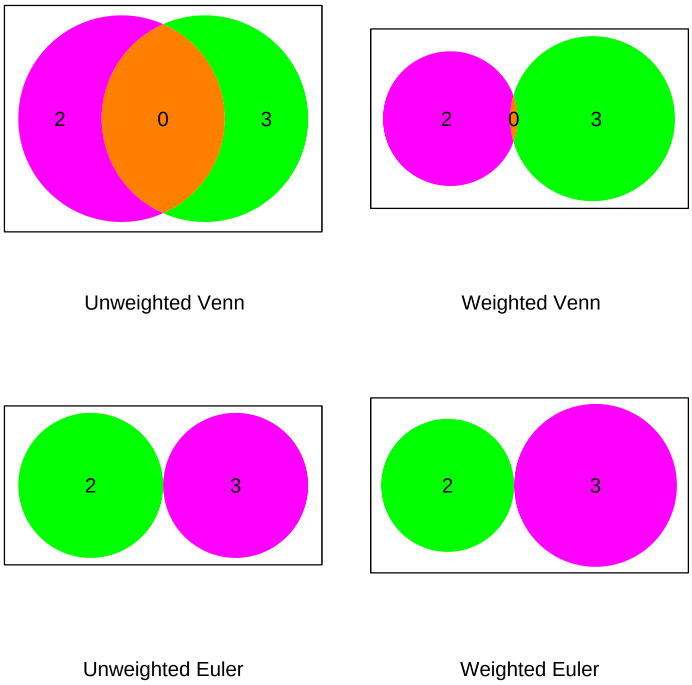
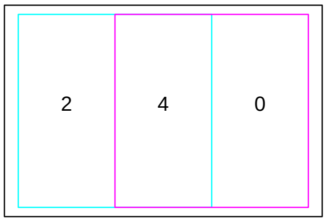


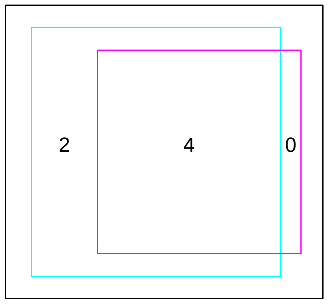
Figure 14: As before for a set of weights with different zeroes

5.1.2 Squares

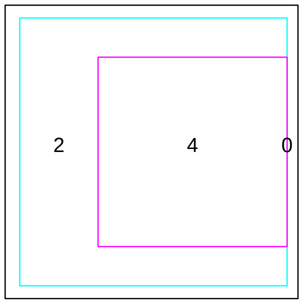
As for circles, the idea of a weighted Venn diagram when some of the weights are zero doesn't make much sense in theory but might be useful for making visual points.



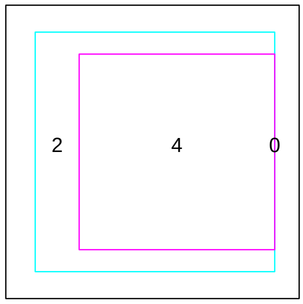
Unweighted Venn



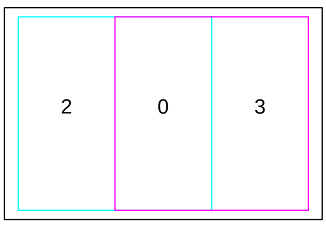
Weighted Venn



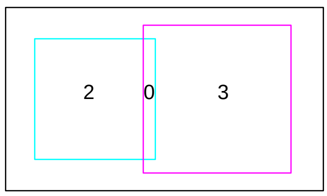
Unweighted Euler



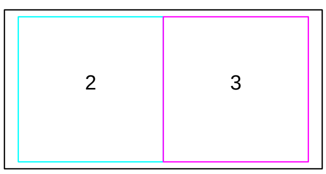
Weighted Euler



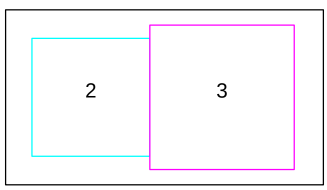
Unweighted Venn



Weighted Venn



Unweighted Euler



Weighted Euler

5.2 3-set Euler diagrams

5.2.1 Circles

There is currently no effect of setting `doEuler=TRUE` for three circles, but the `doWeights=TRUE` flag does an approximate job. There are about 40 distinct ways in which intersection regions can have zeroes can occur, but here are some examples.

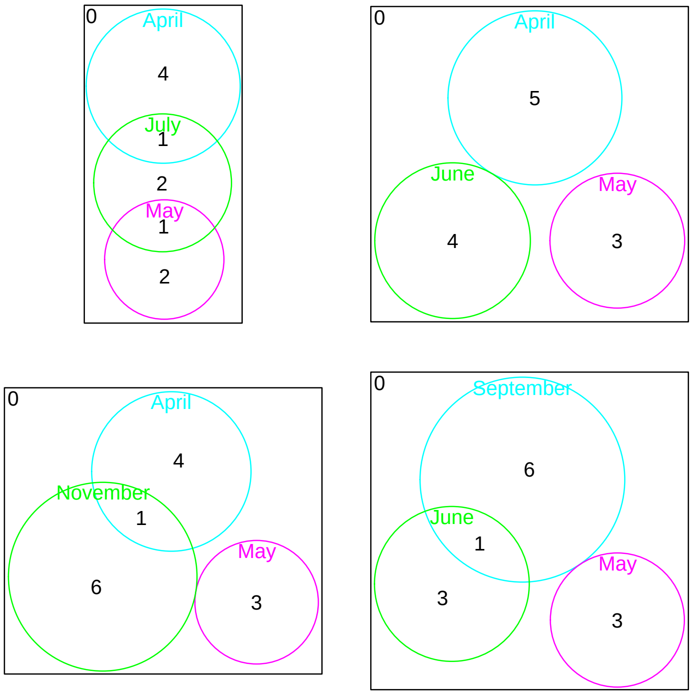


Figure 15: Weighted 3d Venn empty intersections

5.2.2 Triangles

The `doEuler` flag has no effect for triangles; all the weighted diagrams produced are Euler diagrams.

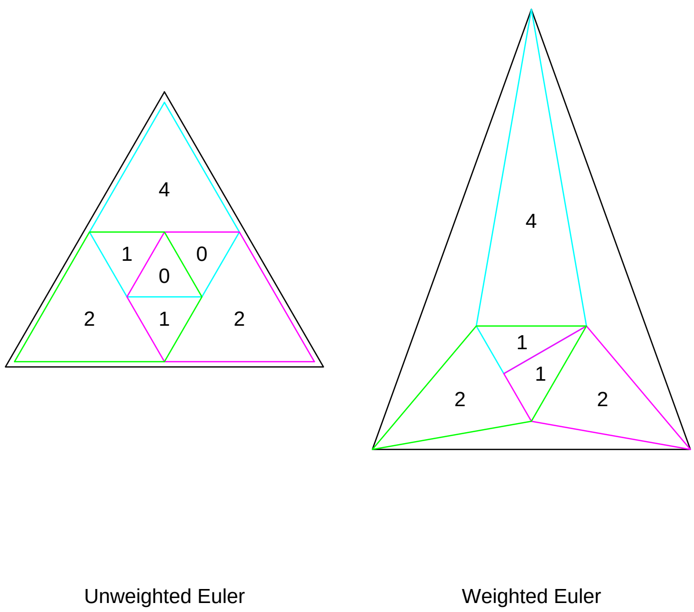


Figure 16: 3d Venn triangular with two zero weights

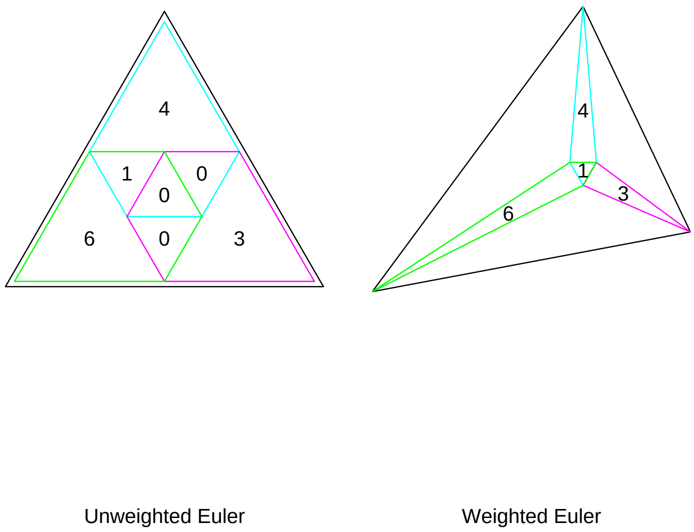


Figure 17: 3d Venn triangular witht three zero weights

5.3 4-set Euler diagrams

5.3.1 Chow-Ruskey diagrams

The `doEuler` flag has no effect for Chow-Ruskey; all the weighted diagrams produced are Euler diagrams.

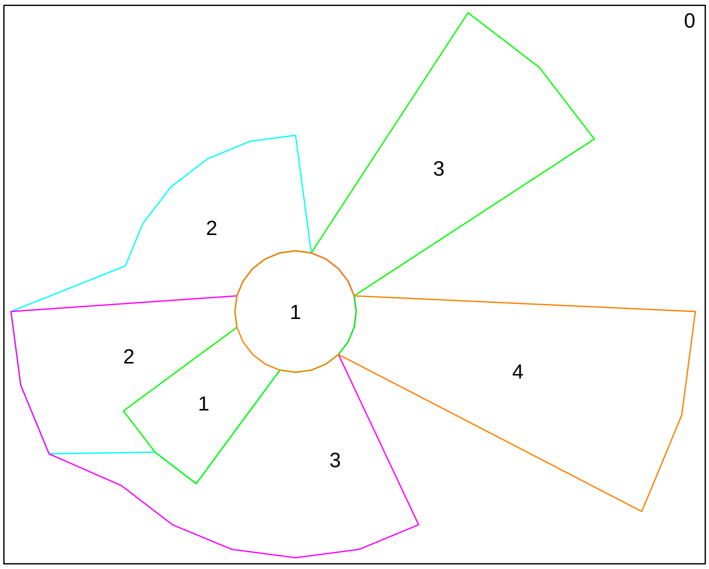


Figure 18: Chow-Ruskey diagram with some zero weights

6 Constructing Venn objects

A Venn object has weights associated with each intersection; by default the weight is the number of elements in the intersection. Venn objects can be constructed with arbitrary weights:

```
> V3 <- Venn(SetNames = month.name[1:3])
> Weights(V3) <- c(0, 81, 81, 9, 81,
+     9, 9, 1)
```

The resulting object has a number of functions defined on it:

```
> SetNames(V3)

[1] "January"  "February" "March"

> Weights(V3)

000 100 010 110 001 101 011 111
  0  81  81   9  81   9   9   1

> NumberOfSets(V3)

[1] 3

> Indicator(V3)

     January February March
[1,]       0        0     0
[2,]       1        0     0
[3,]       0        1     0
[4,]       1        1     0
[5,]       0        0     1
[6,]       1        0     1
[7,]       0        1     1
[8,]       1        1     1
```

6.1 Diagrams for 2 Sets

Venn objects can be restricted to a smaller number of Sets.

```
> V2 <- VN3[, 1:2, ]
> V2 <- VN3[, c("January", "February"),
+     ]
> Weights(V2)

00 10 01 11
 3  2  3  4
```

Note how the weights of the intersections have been updated.

7 Graphics control

TODO rather more here

7.1 Annotation

There are number of annotation options

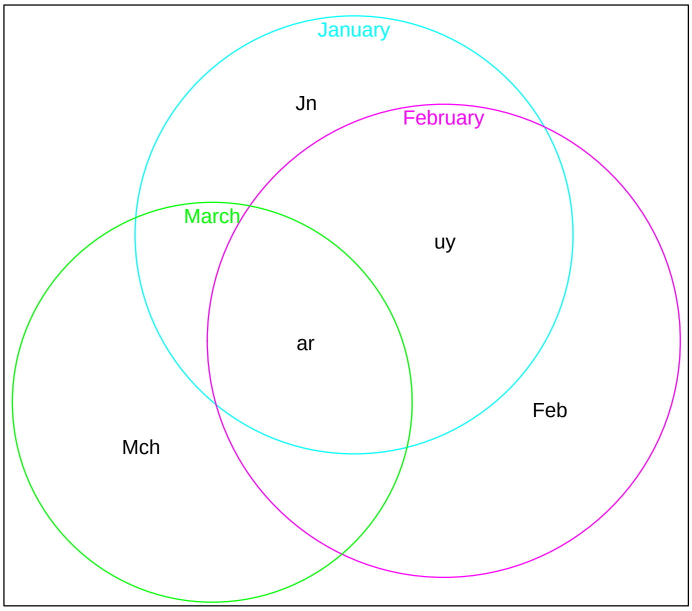


Figure 19: Approximate weighted 3d Venn showing element set membership

7.2 Filled sets and transparency

Some geometries allow for a separate colour specification for each intersection region. Even for those that don't, a similar effect can be obtained using transparent fill colours, but this is only implemented for some graphics devices. In particular it cannot be seen within R under Windows, but it is implemented with PDF devices with version number greater than 1.4.

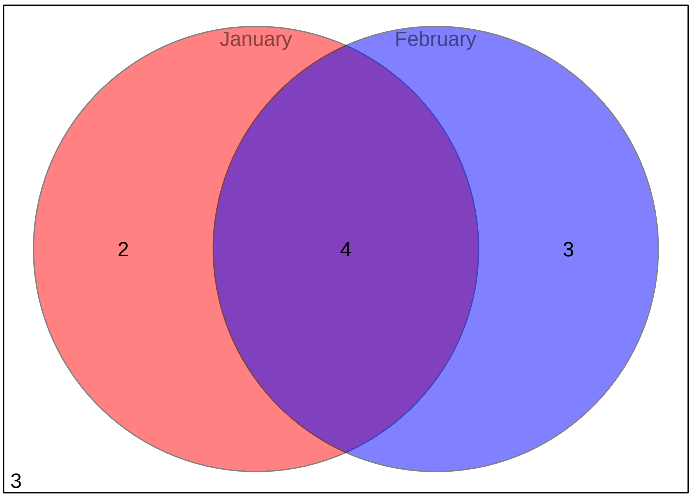


Figure 20: 2D Venn diagram with transparency

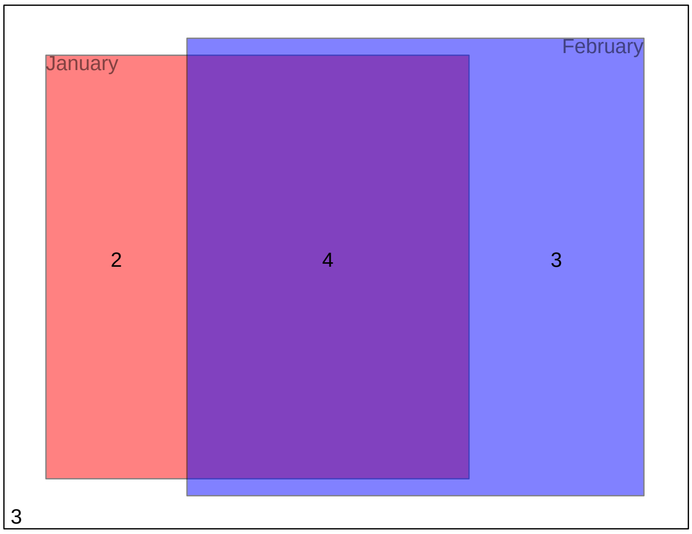


Figure 21: 2D square Venn diagram with transparency

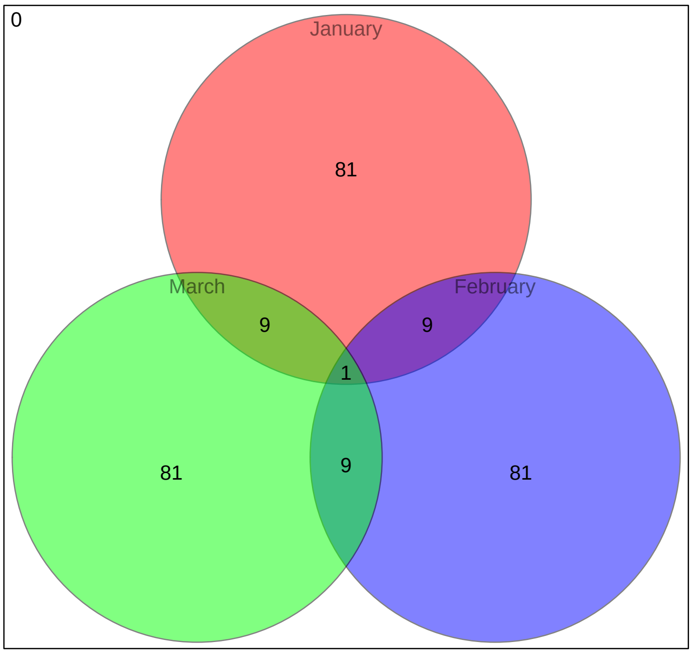


Figure 22: 3D Venn diagram with transparency

8 This document

Author	Jonathan Swinton
CVS id of this document	Id: Venn.Rnw,v 1.24 2007/04/23 22:10:13 js229 Exp .
Generated on	19^{th} June, 2007
R version	R version 2.6.0 Under development (unstable) (2007-06-11 r41912)

References

[1] Lewis Carroll. *Symbolic Logic*. Macmillan, 1896.

[2] Stirling Chow and Frank Ruskey. Drawing area-proportional Venn and Euler diagrams. In Giuseppe Liotta, editor, *Graph Drawing*, volume 2912 of *Lecture Notes in Computer Science*, pages 466–477. Springer, 2003.

[3] Stirling Chow and Frank Ruskey. Towards a general solution to drawing area-proportional Euler diagrams. *Electr. Notes Theor. Comput. Sci.*, 134:3–18, 2005.

[4] A. W. F. Edwards. *Cogwheels of the Mind: The Story of Venn Diagrams*. The John Hopkins University Press, Baltimore, Maryland, 2004.

[5] Frank Ruskey and Mark Weston. *The Electronic Journal of Combinatorics*, chapter A Survey of Venn Diagrams. Jun 2005.

[6] J. Venn. On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 9:1–18, 1880.