Venn diagrams
Technical details and regression checks

Jonathan Swinton

13th July, 2009

- Try CR for weight=0
- Plot faces for Chow-Ruskey
- General set membership
- implement not showing dark matter eg Fig 1
- Different choices of first and second sets for AWFE
- Add in the equatorial sets for AWFE
- AWFE-book like figures
- naming of weights for triangles
- likesquares argument for triangles
- likesquares argument for 4-squares
- central dark matter
- Comment on triangles
- Comment on AWFE return geometry
- calculate three circle areas correctly
- text boxes
- use grob objects/printing properly
- "Exact" slot mess
- proper data handling:
- choose order;
- cope with missing data including missing zero intersection;
- Define weights via names
- graphical parameters
- discuss Chow-Ruskey zero=nonsimple

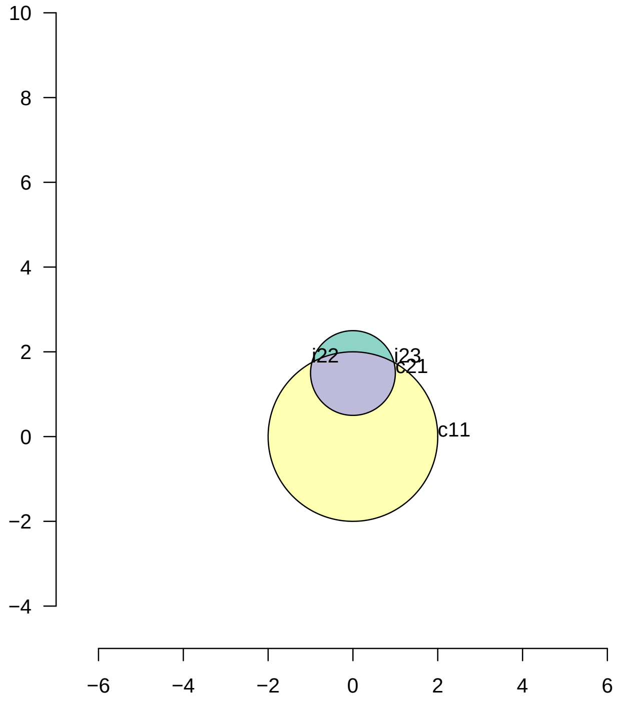
1 Venn objects

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

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

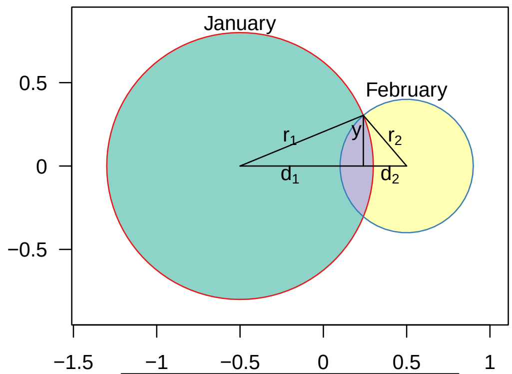
```
> setList <- strsplit(month.name, split = "")
> names(setList) <- month.name
> VN3 <- VennFromSets(setList[1:3])
> V2 <- VN3[, c("January", "February"),
+     ]

> V4 <- VennFromSets(setList[1:4])
> V4f <- V4
> V4f@IndicatorWeight[, ".Weight"] <- 1

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

> V3.big <- Venn(SetNames = month.name[1:3],
+     Weight = 2^(1:8))
> V2.big <- V3.big[, c(1:2)]

> Vempty <- VennFromSets(setList[c(4,
+     5, 7)])
> Vempty2 <- VennFromSets(setList[c(4,
+     5, 11)])
> Vempty3 <- VennFromSets(setList[c(4,
+     5, 6)])
```

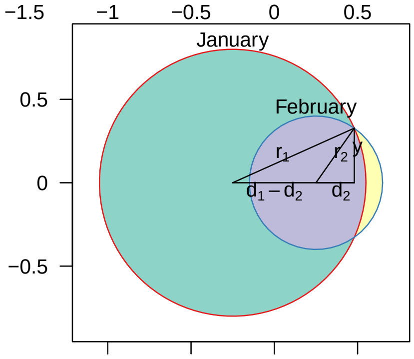
2 The VennDrawing object



3 Two circles

3.1 Two circles





There is an intersection if $|r_1 - r_2| < d < r_1 + r_2$. If so and $d < \max(r_1, r_2$ the centre of the smaller circle is in the interior of the larger.

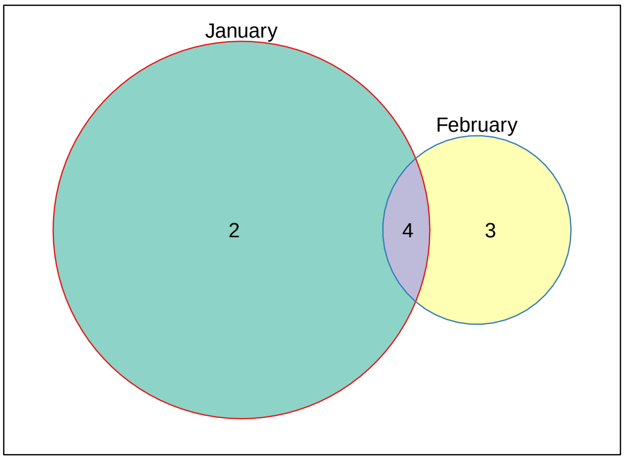
We rely on the relationships

$$\begin{aligned} d_1^2 + y^2 &= r_1^2 \\ d_2^2 + y^2 &= r_2^2 \end{aligned}$$

If $\max(r_1, r_2) < d < r_1 + r_2$ then $d = d_1 + d_2$; if $|r_1 - r_2| < d < \max(r_1, r_2) then$d=|d$_1 - d_2|$.

We rely on the relationships

$$\begin{aligned} d_1 &= (d^2 - r_2^2 + r_1^2)/(2d) \\ d_2 &= |d - d_1| \\ y &= frac12d\sqrt{4d^2 r_1^2 - (d^2 - r_2^2 + r_1^2)^2} \\ &= \sqrt{r_1^2 - d_1^2} \end{aligned}$$


January
February
2
4
3

3.2 Weighted 2-set Venn diagrams for 2 Sets

3.2.1 Circles

It is always possible to get an exactly area-weighted solution for two circles as shown in Figure 1.

```
      00        11        10        01
475.9979 271.9995  67.9992 135.9992
```

```
[1] Area            Weight
[3] IndicatorString Density
<0 rows> (or 0-length row.names)
```

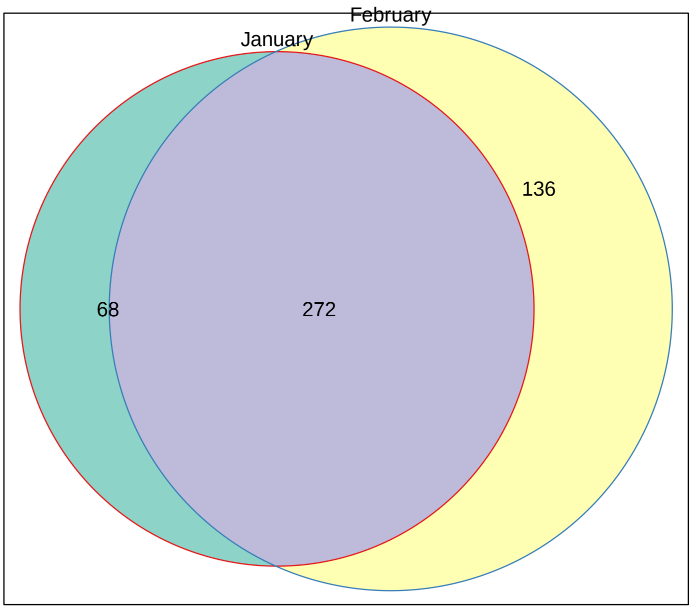


Figure 1: Weighted 2d Venn

3.3 2-set Euler diagrams

3.3.1 Circles

```
       00         11         10         01
7.1339724 3.8633868 0.1352894 3.1352961
```

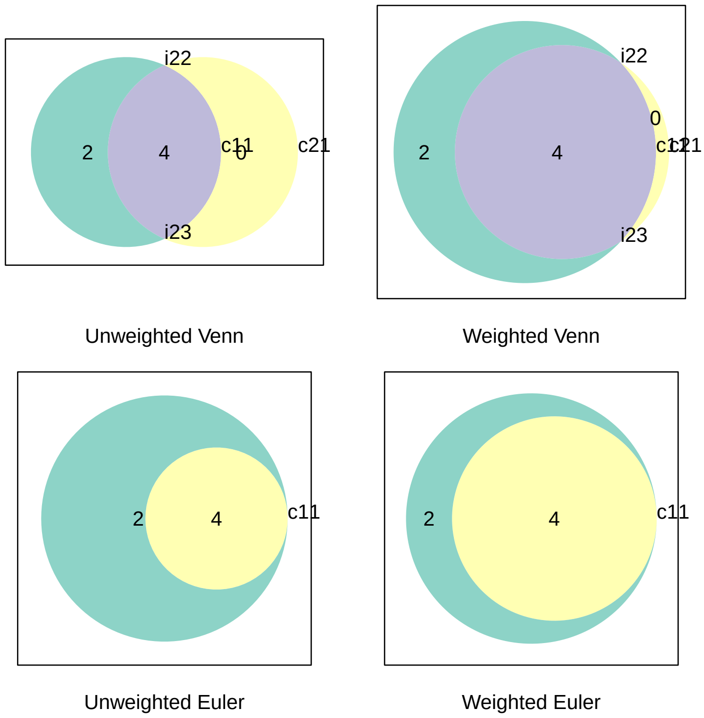


Figure 2: Effect of the Euler and `doWeights` flags.

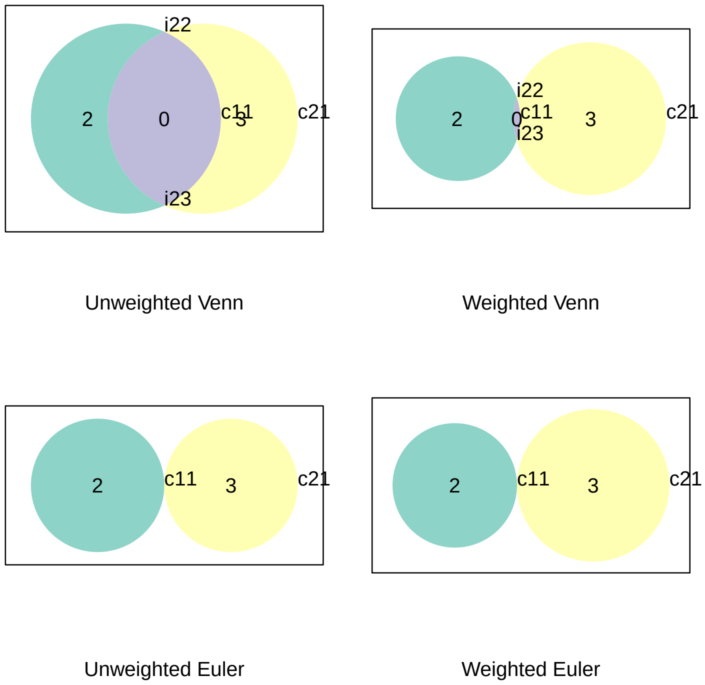


Figure 3: As before for a different set of weights

w=compute.C2(V=V2.no10,doEuler=TRUE,doWeights=FALSE)

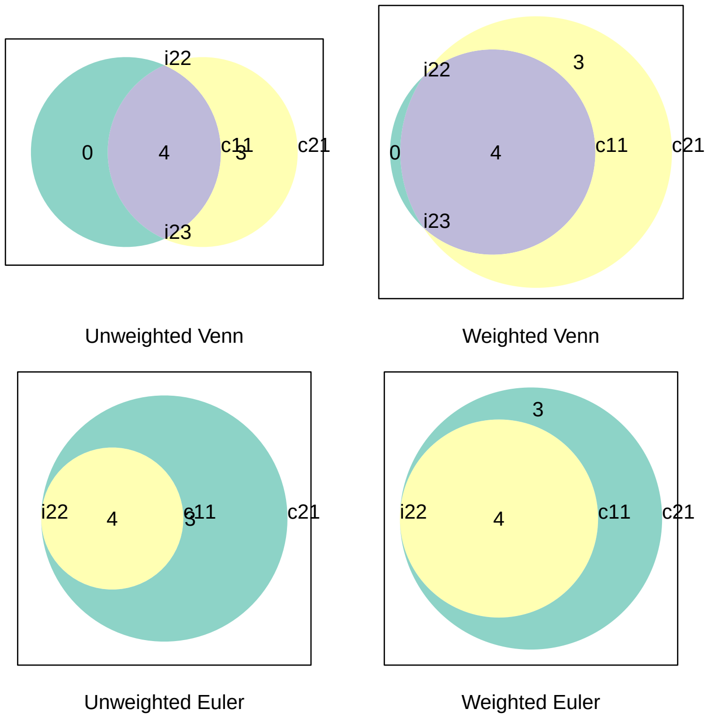
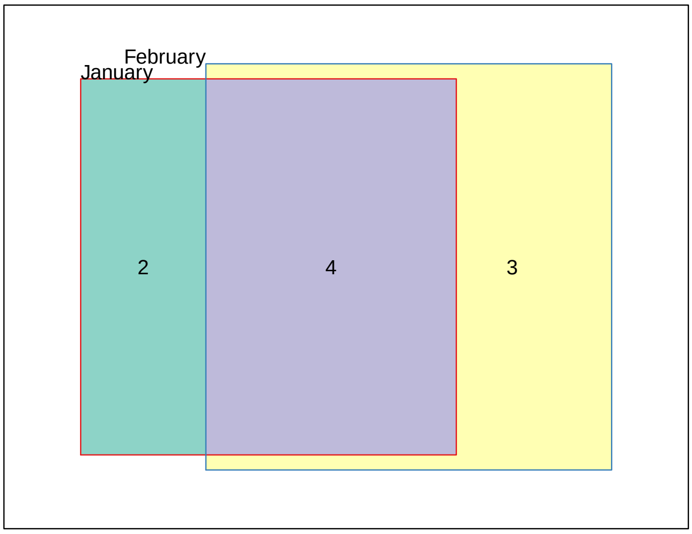


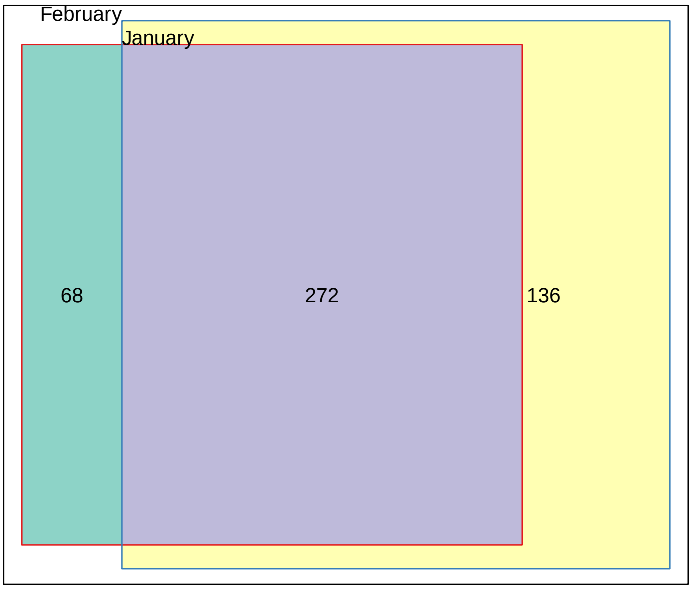
Figure 4: As before for a different set of weights

4 Two squares

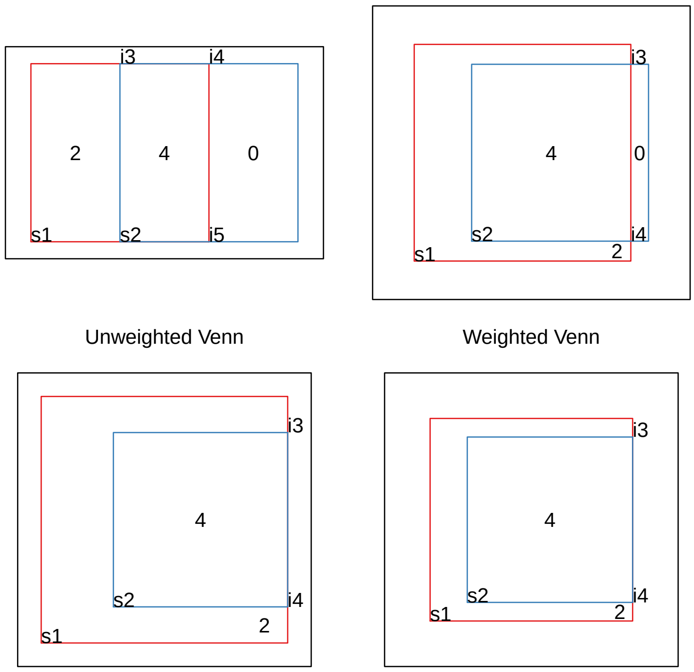


4.0.2 Weights

```
 00  11  10  01
476 272  68 136
```



4.0.3 Squares

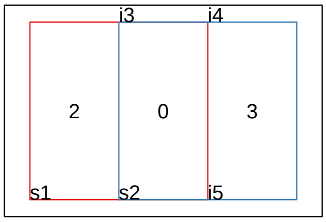


Unweighted Venn

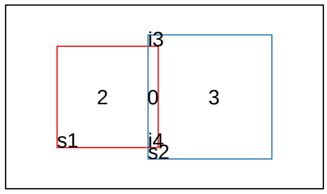
Weighted Venn

Unweighted Euler

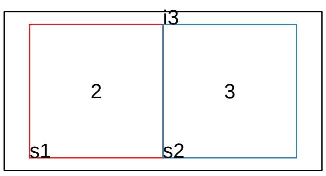
Weighted Euler



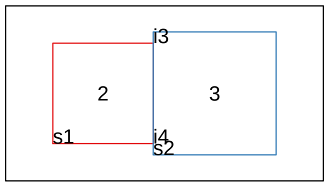
Unweighted Venn



Weighted Venn

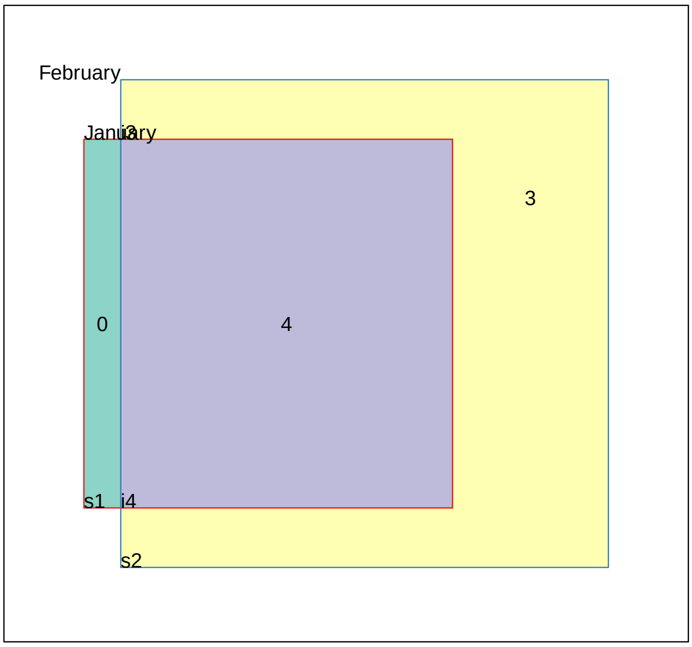


Unweighted Euler



Weighted Euler

```
 00  11  10  01
7.4 3.6 0.4 3.4
```



5 Three circles

```
> plot(Vcombo, doWeights = FALSE, show = list(Faces = TRUE))
```

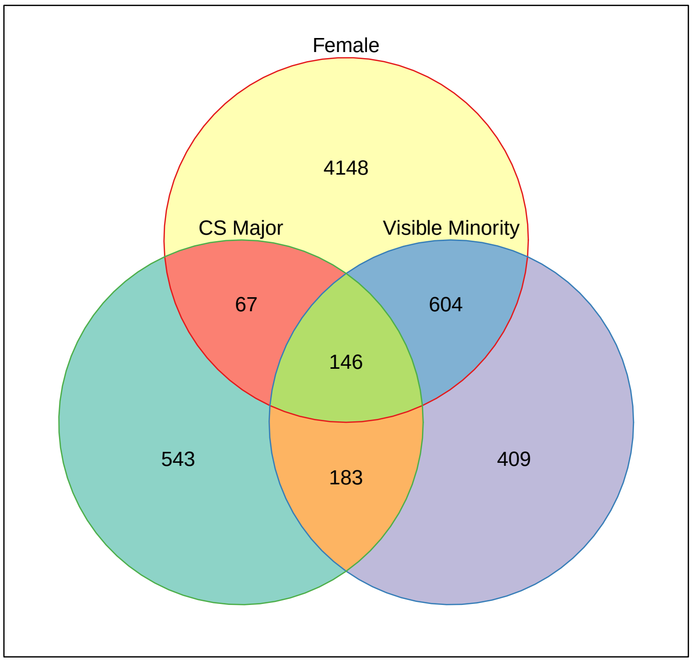


Figure 5: A three-circle Venn diagram

5.0.4 Weights

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

```
        000         001         101         100
6094.83358  537.83535   72.16413 4142.83542
        111         110         011         010
 140.83530  609.16384  188.16391  403.83563

[1] Area            Weight
[3] IndicatorString Density
<0 rows> (or 0-length row.names)
```

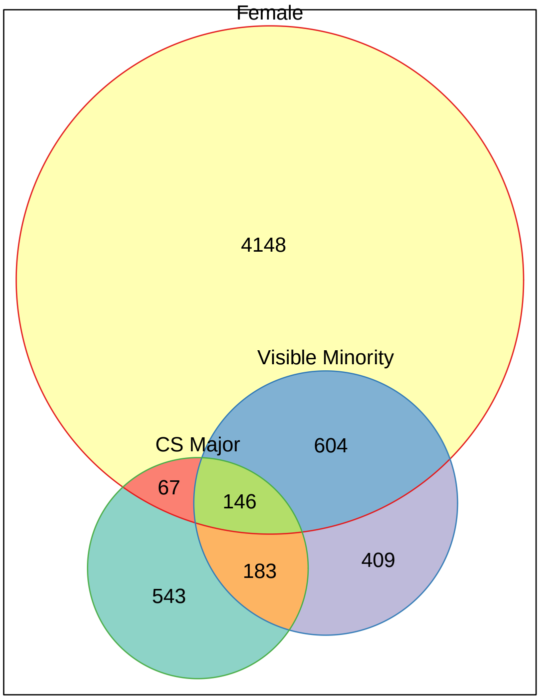


Figure 6: 3D Venn diagram. All of the areas are correct to within 10%

6 Three Triangles

The triangular Venn diagram on 3-sets lends itself nicely to an area-proportional drawing under some contrainsts on the weights

```
[1] Area            Weight
[3] IndicatorString Density
<0 rows> (or 0-length row.names)
```

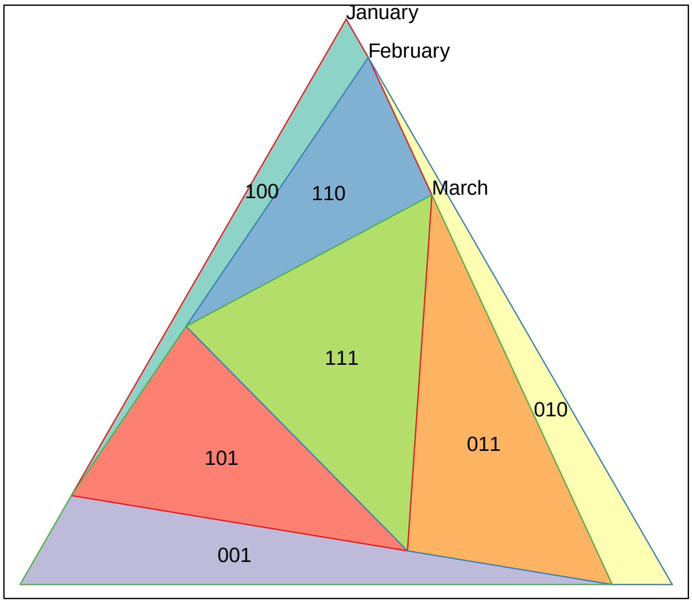


Figure 7: Triangular Venn with external universe

6.1 Triangular Venn diagrams

6.1.1 Triangles

```
000  100  010  111  110  001
 12    2    3    2    2    3
```

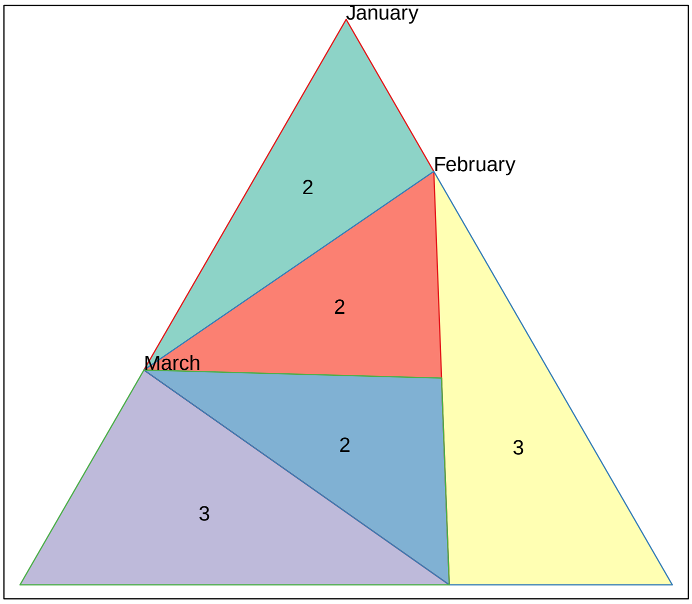


Figure 8: 3d Venn triangular with one empty intersection

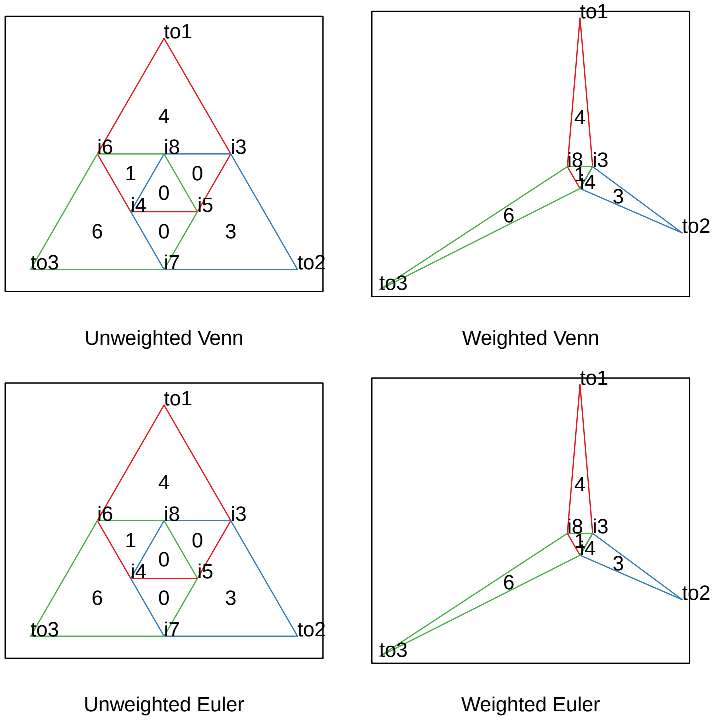
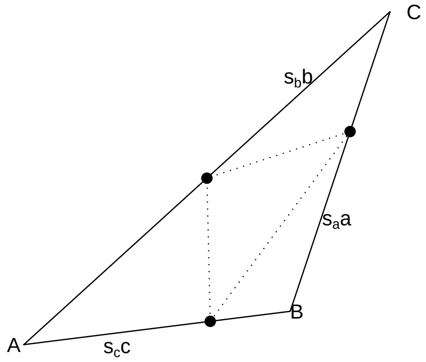


Figure 9: 3d Venn triangular with two empty intersection



Given a triangle ABC of area Δ and some nonnegative weights $w_a + w_b + w_c < 1$ we want to set s_c, s_a and s_b so that the areas of each of the apical triangles are Δ-proportional to w_a, w_b and w_c. This means

$$s_c(1-s_b)bc\sin A = 2w_a\Delta \tag{1}$$
$$s_a(1-s_c)ca\sin B = 2w_b\Delta \tag{2}$$
$$s_b(1-s_a)ab\sin C = 2w_c\Delta \tag{3}$$

So

$$s_c(1-s_b) = w_a \tag{4}$$
$$s_a(1-s_c) = w_b \tag{5}$$
$$s_b(1-s_a) = w_c \tag{6}$$

$$s_b = 1 - w_a/s_c \tag{7}$$
$$s_a = w_b/(1-s_c) \tag{8}$$
$$(s_c - w_a)(1 - s_c - w_b) = s_c(1-s_c)w_c \tag{9}$$

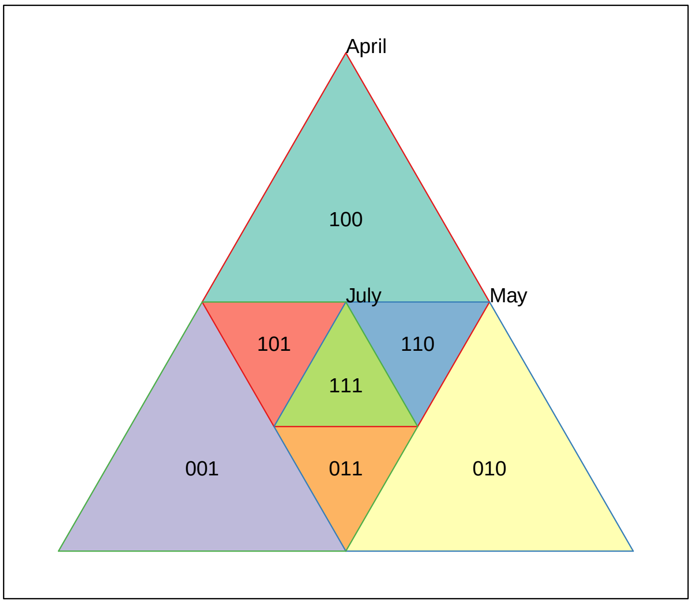
$$s_c^2(1-w_c) + s_c(w_b + w_c - w_a - 1) + w_a(1-w_b) = 0 \tag{10}$$

Iff

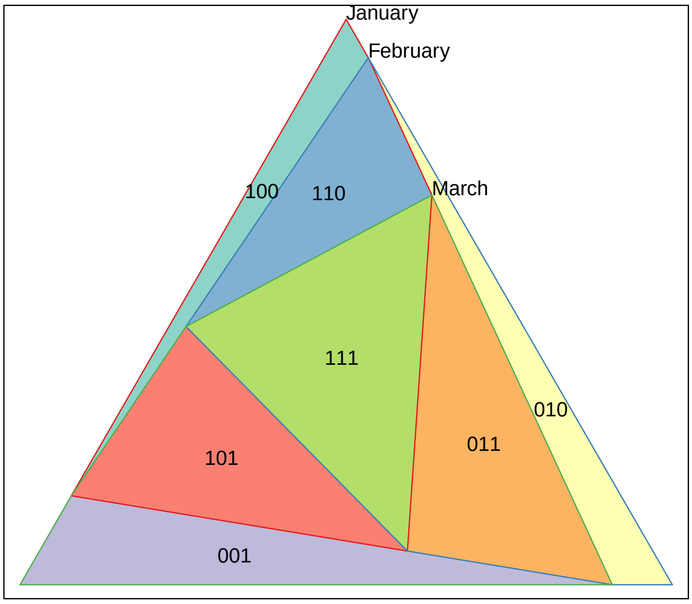
$$4w_a w_b w_c < (1-(w_a+w_b+w_c))^2 \tag{11}$$

this has two real solutions between w_a and $1-w_b$.

```
[1] TRUE
```



6.2 Three triangles



7 Three Squares

This is a version of the algorithm suggested by **?**]. TODO likesquares

```
[1] Area            Weight
[3] IndicatorString Density
<0 rows> (or 0-length row.names)
```

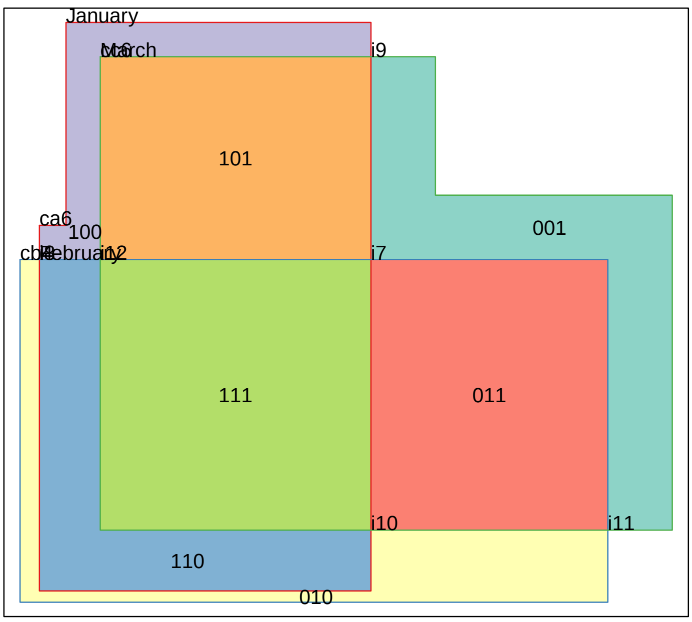
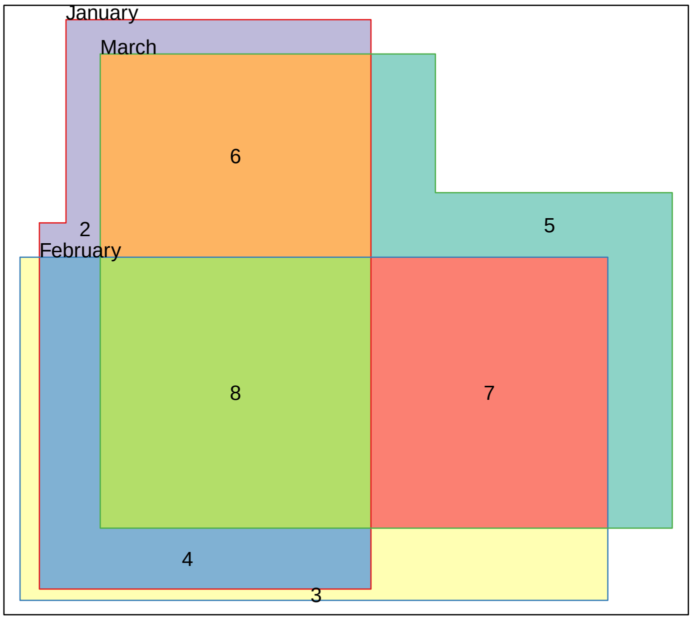


Figure 10: Weighted 3-set Venn diagram based on the algorithm of **?**]

7.1 Three squares



8 Four squares

8.1 Unweighted 4-set Venn diagrams

```
> doans <- function(V4, s, likeSquares) {
+     S4 <- compute.S4(V4, s = s, likeSquares = likeSquares)
+     CreateViewport(S4)
+     PlotSetBoundaries(S4, gp = gpar(lwd = 4:1,
+         col = trellis.par.get("superpose.symbol")$col))
+     UpViewports()
+ }
```

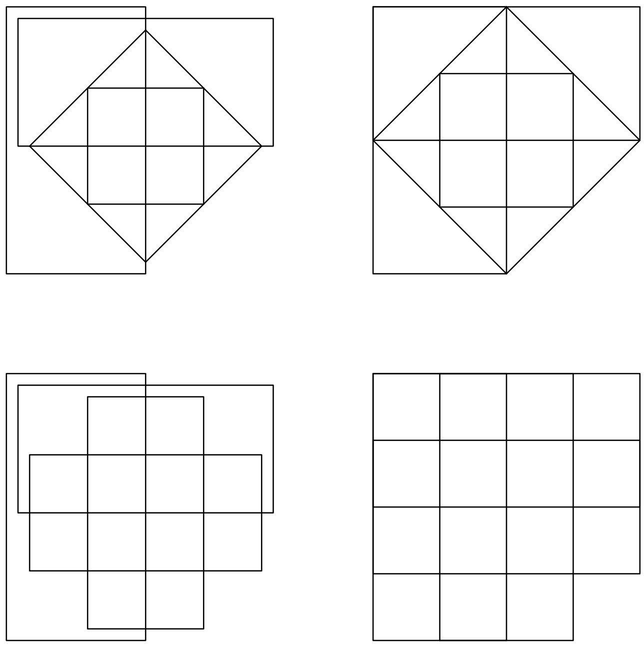
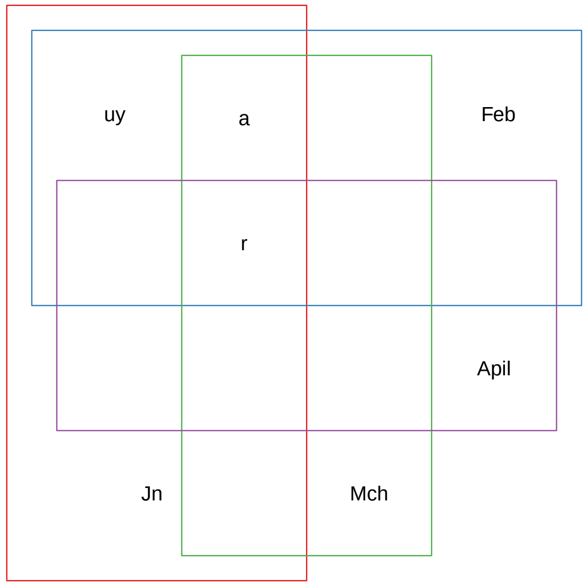
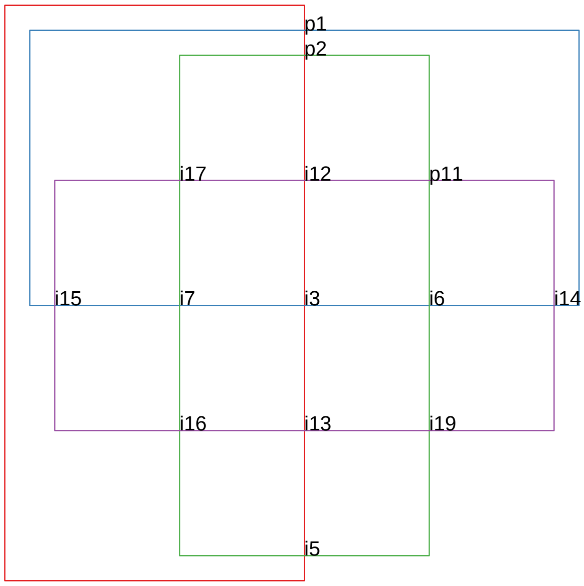


Figure 11: Four variants on the four-squares

8.2 Four squares




p1
p2
i17
i12
p11
i15
i7
i3
i6
i14
i16
i13
i19
i5

9 Four Ellipses

Ellipses don’t have faces or nodes, and can’t have weights sent.

DOES NOT WORK

10 Chow-Ruskey

See [? ?].

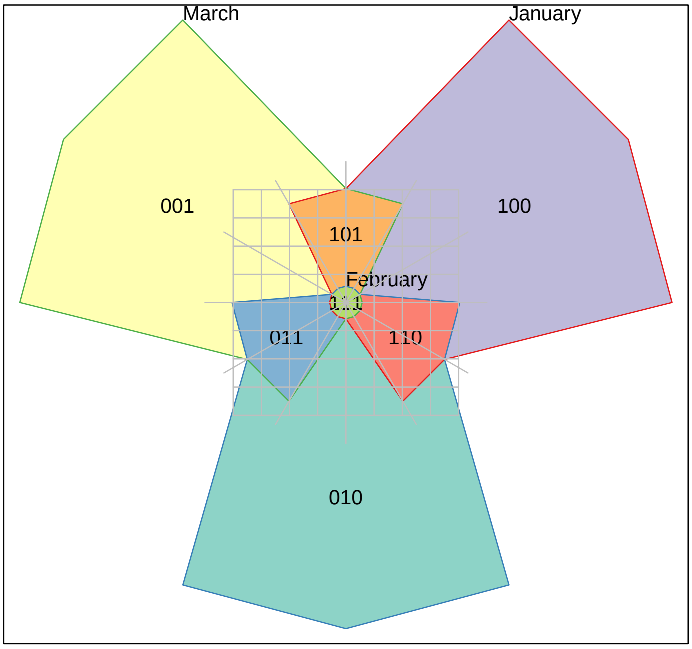
10.1 Chow-Ruskey diagrams for 3 sets

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

```
[1] Area            Weight
[3] IndicatorString Density
<0 rows> (or 0-length row.names)
```

Figure 12: Chow-Ruskey weighted 3-set diagram

```
[1] Area            Weight
[3] IndicatorString Density
<0 rows> (or 0-length row.names)
```



```
[1] Area            Weight
[3] IndicatorString Density
<0 rows> (or 0-length row.names)
```

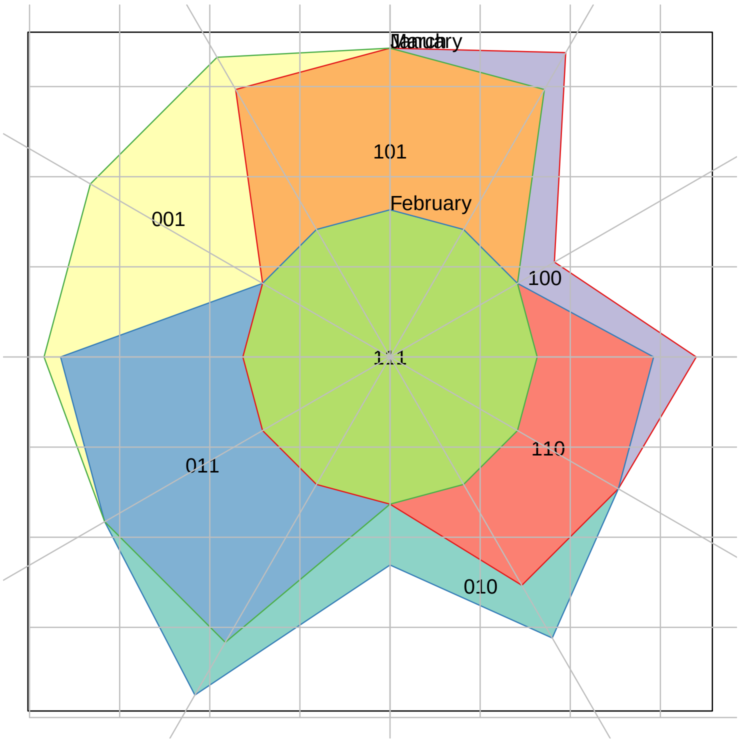


Figure 13: Chow-Ruskey CR3f

10.2 Chow-Ruskey diagrams for 4 sets

```
[1] Area            Weight
[3] IndicatorString Density
<0 rows> (or 0-length row.names)
```

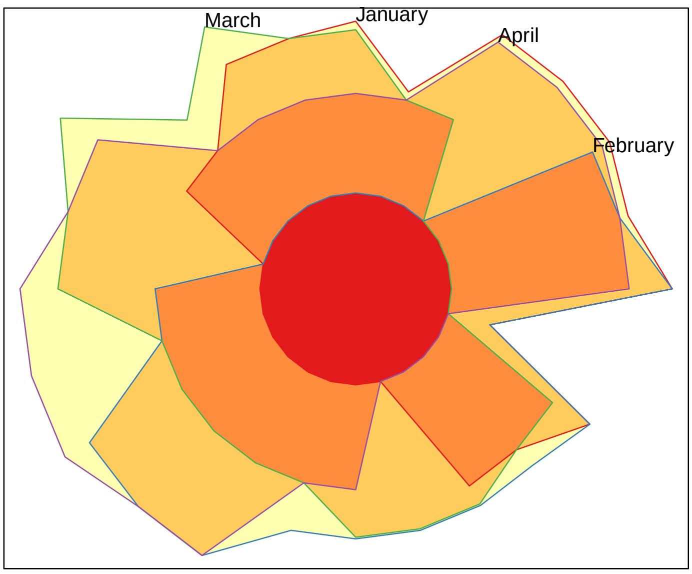


Figure 14: Chow-Ruskey weighted 4-set diagram

```
[1] Area            Weight
[3] IndicatorString Density
<0 rows> (or 0-length row.names)
```

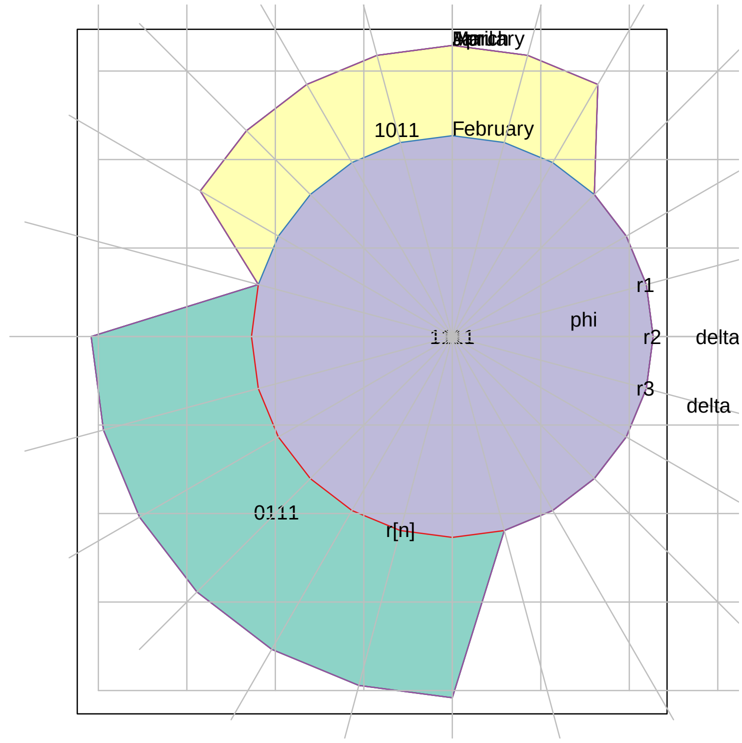


Figure 15: Chow-Ruskey weighted 4-set diagram

The area of the sector $0r_1r_2$ is $\frac{1}{2}r_1r_2\sin\phi$. The area of $0r_1s_2$ is $\frac{1}{2}(r_1(r_2+\delta)\sin\phi)$ and so the area of $r_1r_2s_2$ is $\frac{1}{2}(r_1\delta\sin\phi)$.

The area of $r_2r_2s_2s_3$ is $\frac{1}{2}[(r_3+\delta)(r_2+\delta)-r_3r_2)\sin\phi = \frac{1}{2}[(r_3+r_2)\delta+\delta^2]\sin\phi$.

The total area of the outer shape is

$$
\begin{aligned}
A &= \frac{1}{2}(\sin\phi)\left[(r_1+r_n)\delta+\sum_{k=2}^{n-2}[(r_{k+1}+r_k)\delta+\delta^2]\right] &(12)\\
&= \frac{1}{2}(\sin\phi)\left[(r_1+r_n)\delta+(n-2)\delta^2+\delta\sum_{k=2}^{n-2}[(r_{k+1}+r_k)]\right] &(13)\\
&= \frac{1}{2}(\sin\phi)\left[(r_1+r_2+2r_3+\ldots+2r_{n-2}+r_{n-1}+r_n)\delta+(n-3)\delta^2\right] &(14)
\end{aligned}
$$

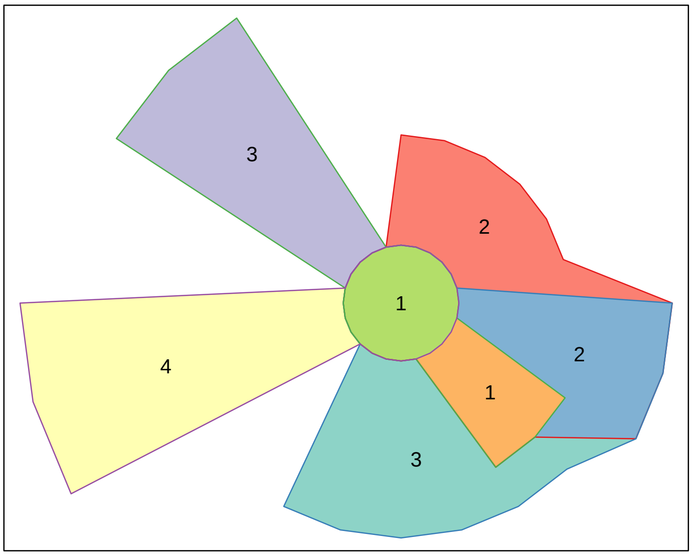
so

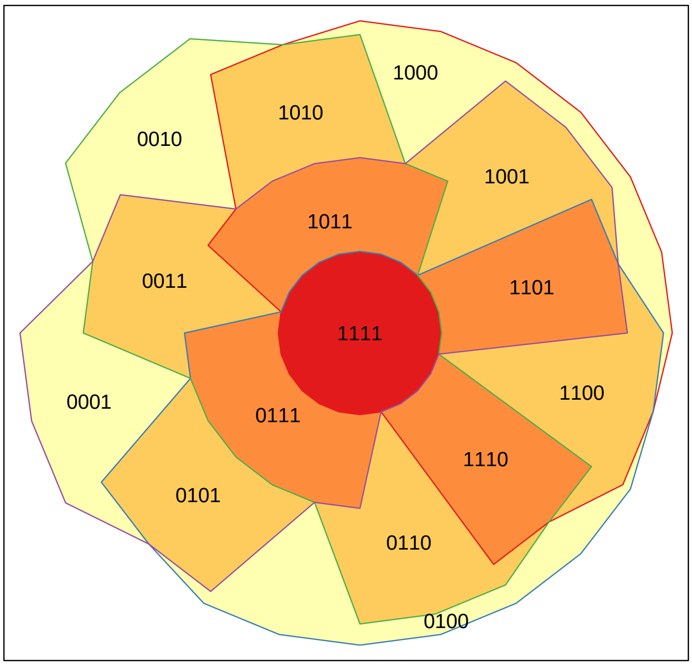
$$
\begin{aligned}
0 &= c_a\delta^2+c_b\delta+c_c &(15)\\
c_a &= n-3 &(16)\\
c_b &= r_1+r_2+2r_3+\ldots+2r_{n-2}+r_{n-1}+r_n &(17)\\
c_c &= -A/\frac{1}{2}\sin\phi &(18)
\end{aligned}
$$

This is implemented in the compute.delta function.

If all the rs are the same then $c_b = [2(n-3)+4]r = (2n-2)r$.

```
[1] Area            Weight
[3] IndicatorString Density
<0 rows> (or 0-length row.names)
```




1000
1010
0010
1001
1011
0011
1101
1111
1100
0001
0111
1110
0101
0110
0100

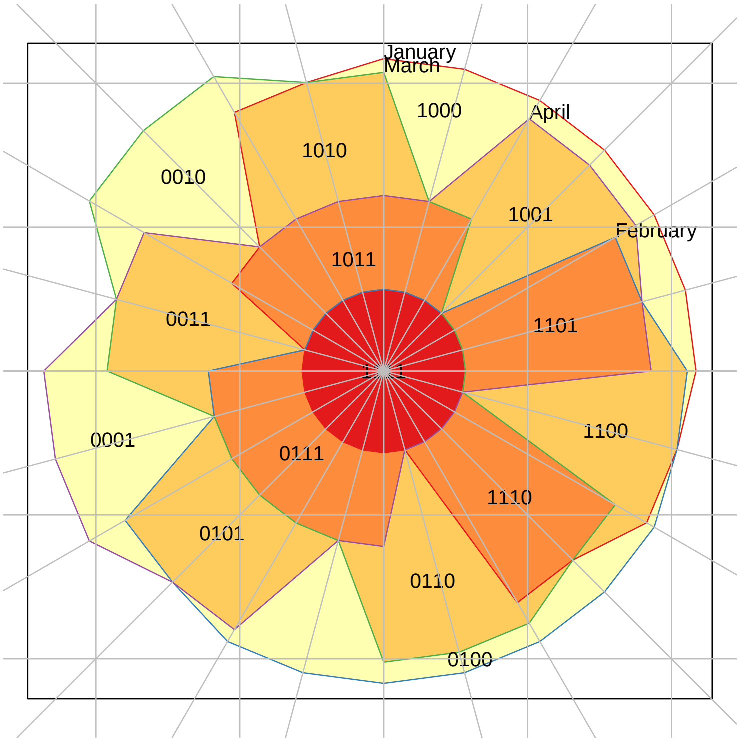


Figure 16: Chow-Ruskey 4

11 Euler diagrams

11.1 3-set Euler diagrams

11.1.1 Other examples of circles

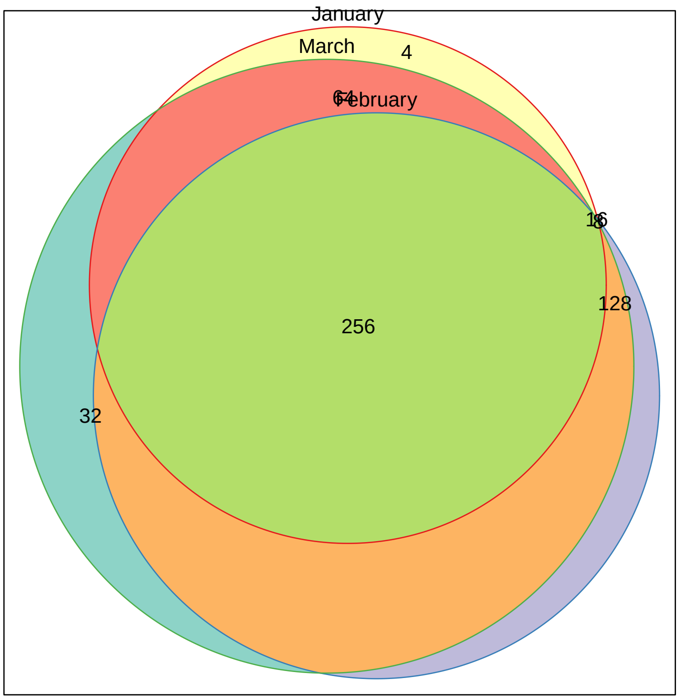


Figure 17: TODO Big weighted 3d Venn fails

12 Error checking

These should fail

```
> print(try(Venn(numberOfSets = 3, Weight = 1:7)))

[1] "Error in Venn(numberOfSets = 3, Weight = 1:7) : \n  Weight length does not match numb
attr(,"class")
[1] "try-error"

> print(try(V3[1, ]))

[1] "Error in V3[1, ] : Can't subset on rows\n"
attr(,"class")
[1] "try-error"
```

Empty objects don't work

```
character(0)
```

13 This document

Author	Jonathan Swinton
SVN id of this document	Id: VennDrawingTest.Rnw 8 2009-07-13 20:50:37Z js229 .
Generated on	13^{th} July, 2009
R version	R version 2.9.0 (2009-04-17)