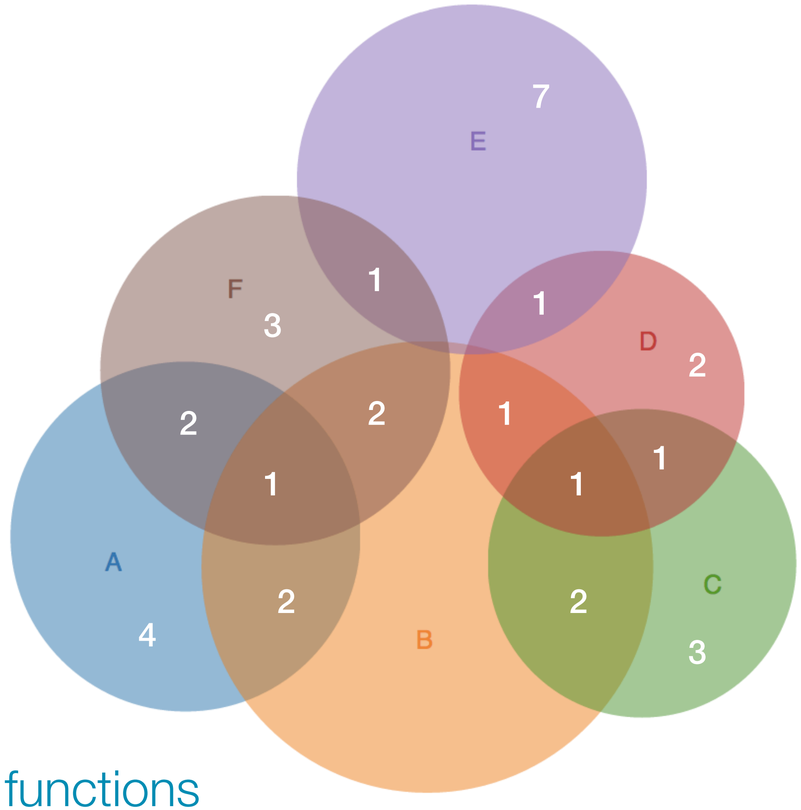
Area-Proportional Venn* Diagrams for D3



New library functions

`d3.layout.venn`	Produces data to draw an area-proportional Venn/Euler diagram.
`venn.size`	Get or set the width/height of the visualization.
`venn.stress`	Retrieve the stress (goodness-of-fit measure) of the Venn diagram.

Data input options

Array of arrays	Each item occupies one row, which contains the groups that it belongs to. Example: [[A_1, A_2, ... A_n], [B_1, B_2, ...B_n], ..., [K_1, K_2, ...K_n]]
1D array	Each element represents the relative size or count of one disjoint region. Note that |A| refers to the number of items belonging to ONLY A. The index is the bitwise OR of the intersecting groups. Example: [|A|, |B|, |A ∩ B|, |C|, |A ∩ C|, |B ∩ C|, |A ∩ B ∩ C|, ...]

Data output

Array of circle objects, each with the following properties:

`x, y`	Center of the circle
`outerRadius`	Size of the circle
`xLabel, yLabel`	Suggested location for a label

Example client use

```
var groups = ["A", "B", "C", "D", "E", "F"];
var data = [];
for (var i=0; i<Math.pow(2, 6); i++)
    data[i] = 0;

var A=0, B=1, C=2, D=3, E=4, F=5;
data[1<<A] = 4; data[1<<B] = 6; data[1<<C] = 3; data[1<<D] = 2;
data[1<<E] = 7; data[1<<F] = 3;
data[1<<A|1<<B] = 2; data[1<<A|1<<F] = 2; data[1<<B|1<<C] = 2;
data[1<<B|1<<D] = 1; data[1<<B|1<<F] = 2; data[1<<C|1<<D] = 1;
data[1<<D|1<<E] = 1; data[1<<E|1<<F] = 1;
data[1<<A|1<<B|1<<F] = 1; data[1<<B|1<<C|1<<D] = 1;

var color = d3.scale.category10();
var venn = d3.layout.venn().size([800, 600]);
var circle = d3.svg.arc().innerRadius(0).startAngle(0).endAngle(2*Math.PI);

var vis = d3.select("body")
  .append("svg")
    .data([data])
    .attr("width", 800).attr("height", 600);

var circles = vis.selectAll("g.arc")
    .data(venn)
  .enter().append("g")
    .attr("class", "arc")
    .attr("transform", function(d, i){
        return "translate(" + d.x + "," + d.y + ")"; });
circles.append("path")
    .attr("fill", function(d, i) { return color(i); })
    .attr("opacity", 0.5)
    .attr("d", circle);
circles.append("text")
    .attr("text-anchor", "middle")
    .text(function(d, i) { return groups[i]; })
    .attr("fill", function(d, i) { return color(i); })
    .attr("x", function(d, i) { return d.labelX; })
    .attr("y", function(d, i) { return d.labelY; });
```

Gradient descent algorithm

It is not generally possible to produce a perfectly accurate Euler diagram when there are >2 sets. This implementation uses a gradient descent algorithm called `venneuler`, which is available as an R package.

Credit:
Leland Wilkinson, “Exact and Approximate Area-proportional Circular Venn and Euler Diagrams.” 2012.
Leland Wilkinson’s `venneuler` source code, available under an MPL license.

*Actually, Euler diagrams, which do not need to contain all possible overlap of sets. Venn diagrams are a subset of Euler diagrams.

Helen Lu, CS448B, Fall 2012