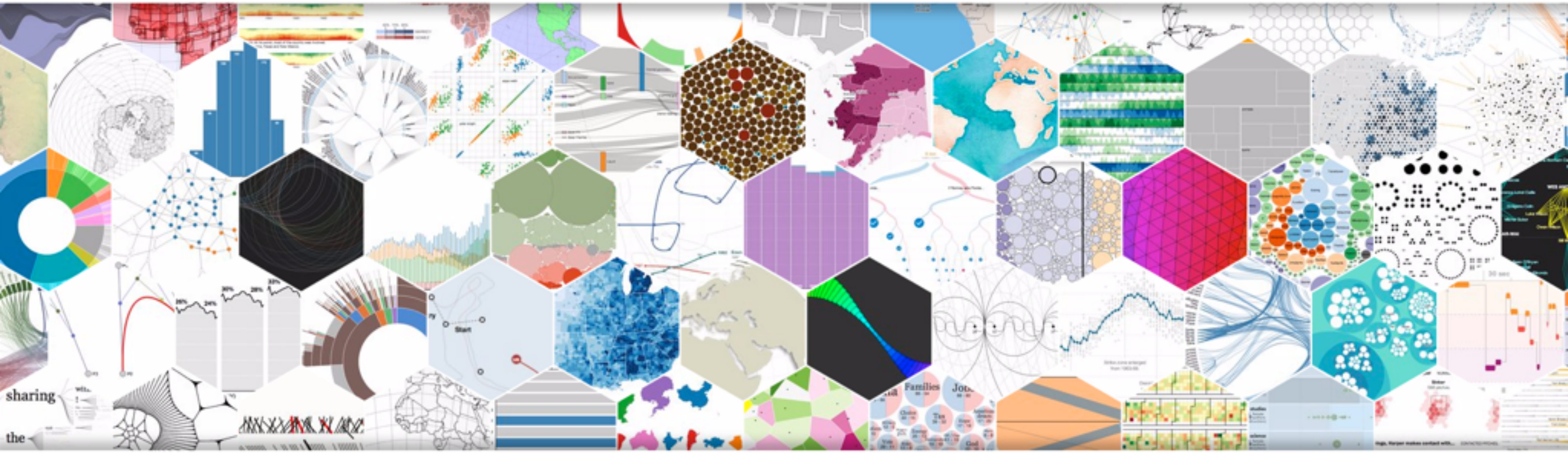
Into the World of migthy D3.js

by @frakti

What D3 is?



*D3 **helps** you bring*
*data to life **using SVG, Canvas** and **HTML**.*

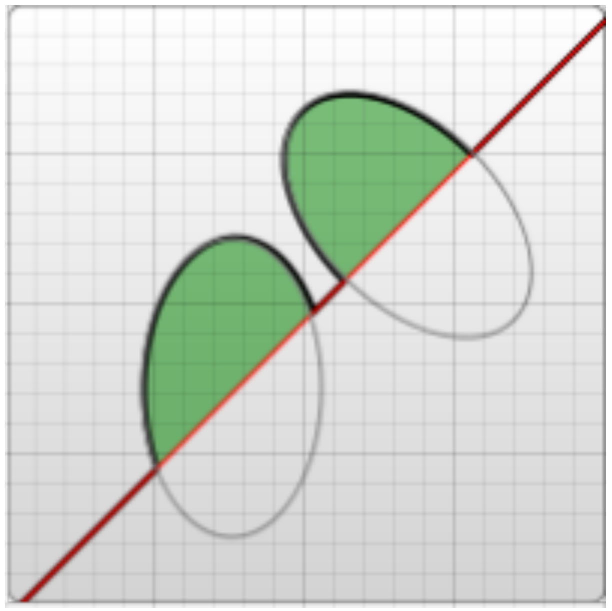
What we can do?

- Literally everything
- Black Box
- *But…*

SVG

- Coordinate system
- Path

```
<path d=„M 10 315
         L 110 215
         A 30 50 0 0 1 162.55 162.45
         L 172.55 152.45
         A 30 50 -45 0 1 215.1 109.9
         L 315 10"
       stroke="black"
       fill="green"
       stroke-width=„2"
       fill-opacity="0.5"/>
```

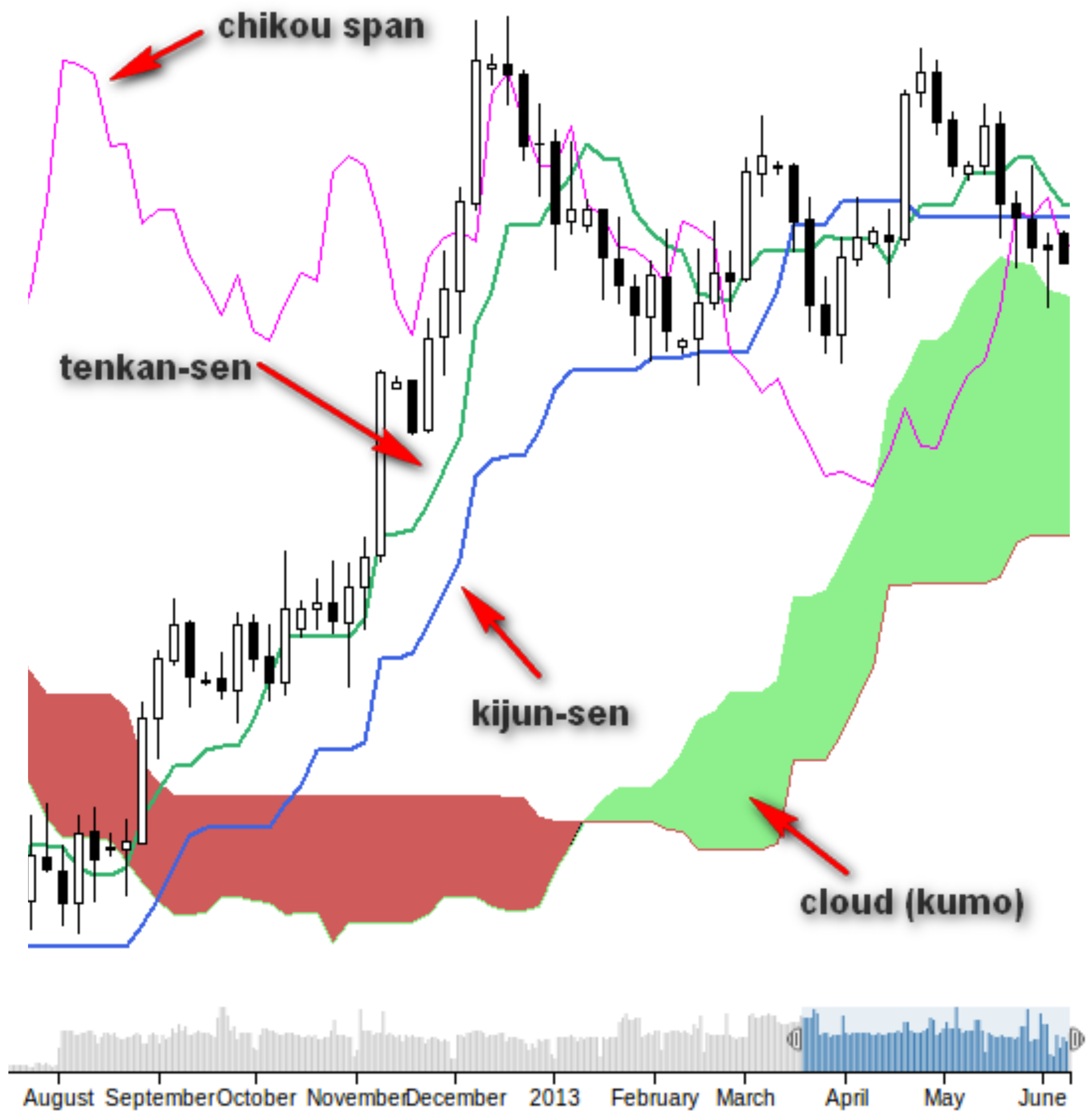


- Clipping and masking
- Styling
- and more (https://www.w3.org/TR/SVG)

Example: Ichimoku*

- SVG
    - path
    - clipPath
    - (rect / polygon)
    - marker
    - text
- D3
    - select
    - scale
    - axis
    - brush
    - area
    - line
    - zoom
    - drag

check **d3fc**, which may help you much building it



D3 modules

Smaller modules first

- d3-collections
  - Nests
- d3-color
- d3-dispath
- **d3-drag**
- **d3-zoom**
- d3-dsv (tsv, csv)
- d3-format
- d3-queue
- d3-random
- d3-request
- d3-time
- d3-time-format
- d3-timer

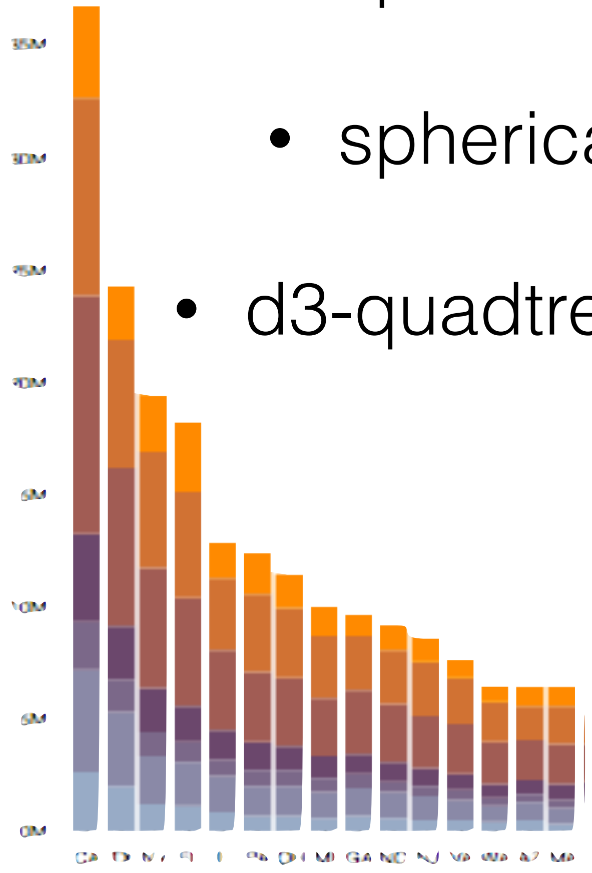
d3-interpolate

Interpolate numbers, colors, strings, arrays, objects, whatever!

Graph modules

Objects

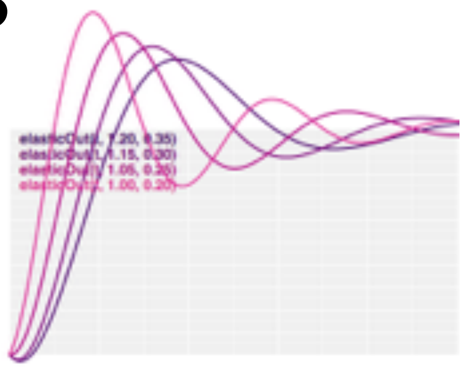
- d3-path
- d3-polygon
- d3-shape
    - arcs
    - pies
    - lines
    - areas
    - curves
    - symbols
    - stacks



GIS

- d3-geo
    - projections
    - spherical shapes
    - spherical math
- d3-quadtree

Animations



- d3-ease
- d3-transition

d3-selection

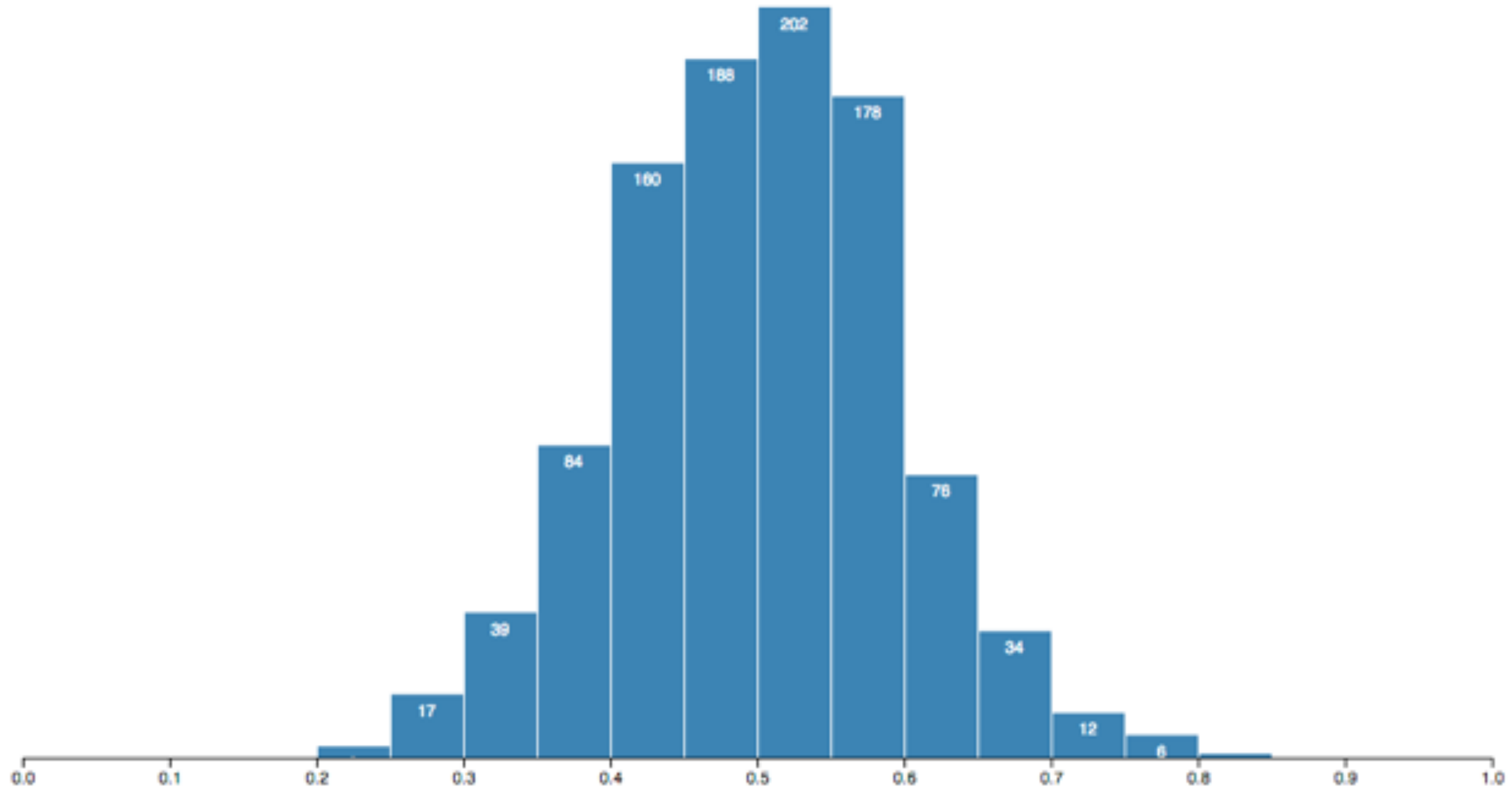
```
// Update…
const p = d3.select("body")
  .selectAll("p")
  .data([{id: 4}, {id: 8}, {id: 15}, {id: 16}, {id: 23}, {id: 42}])
  .text((d) =>  d.id);

// Enter…
p.enter().append("p")
    .text((d) => d.id);

// Exit…
p.exit().remove();
```

d3-arrays

- Statistics
    - extent
- Search
- Transformations
- Histograms



d3-axis

- styling in v3 vs v4

```
var axis = d3.axisLeft(scale)
d3.select("body").append("svg")
    .attr("class", "axis")
    .attr("width", 1440)
    .attr("height", 30)
  .append("g")
    .attr("transform", "translate(0,30)")
    .call(axis);
```

d3-scale

- Continuous Scales (Linear, Power, Log, Identity, Time)
- Sequential Scales
- Quantize Scales
- Ordinal Scales (Band, Point, Category)

- domain
- range

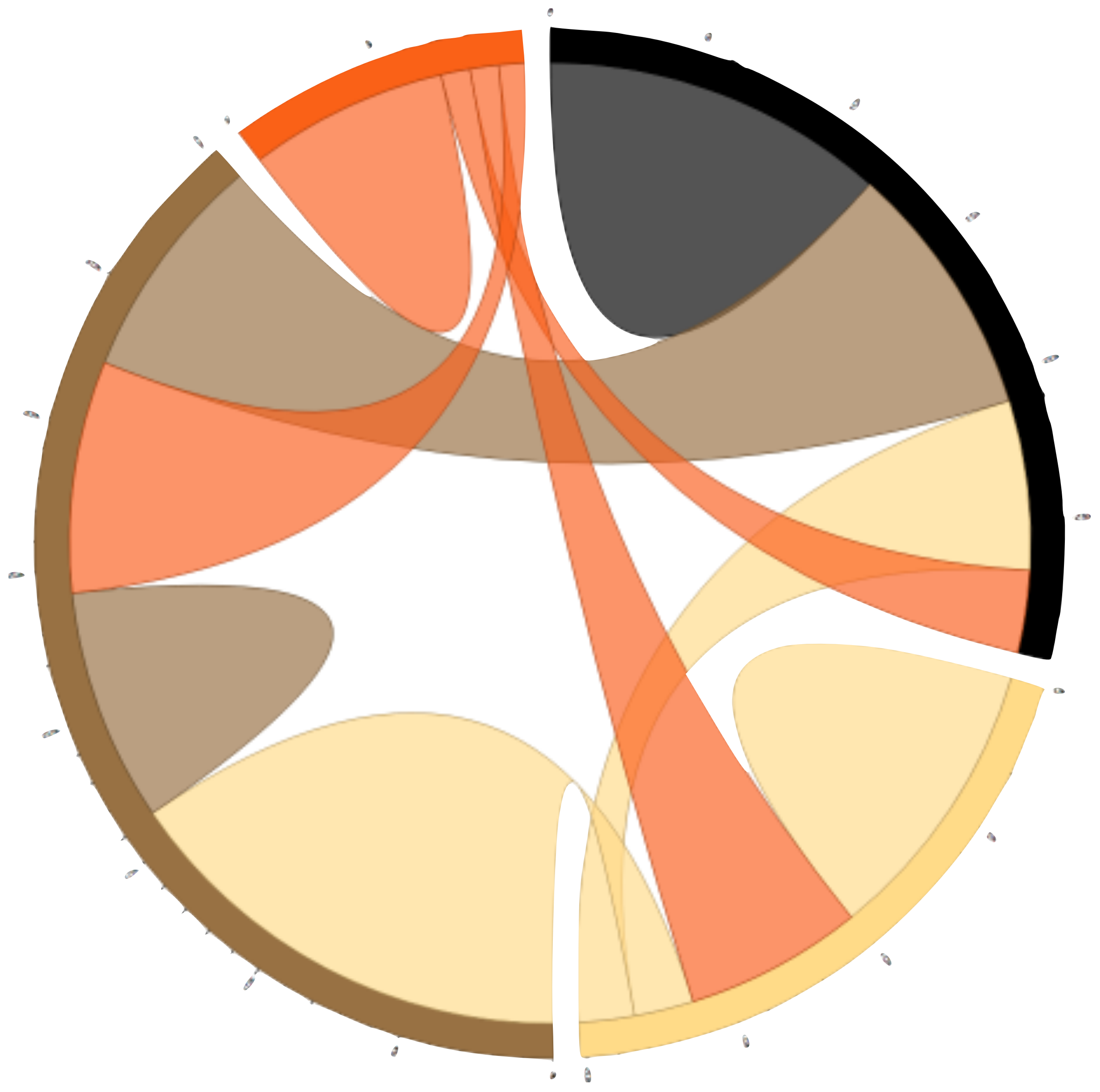
```
var color = d3.scaleLinear()
    .domain([-1, 0, 1])
    .range(["red", "white", "green"]);

color(-0.5); // "rgb(255, 128, 128)"
color(+0.5); // "rgb(128, 192, 128)"
```

d3-brush

Select a one- or two-dimensional region using the mouse or touch.

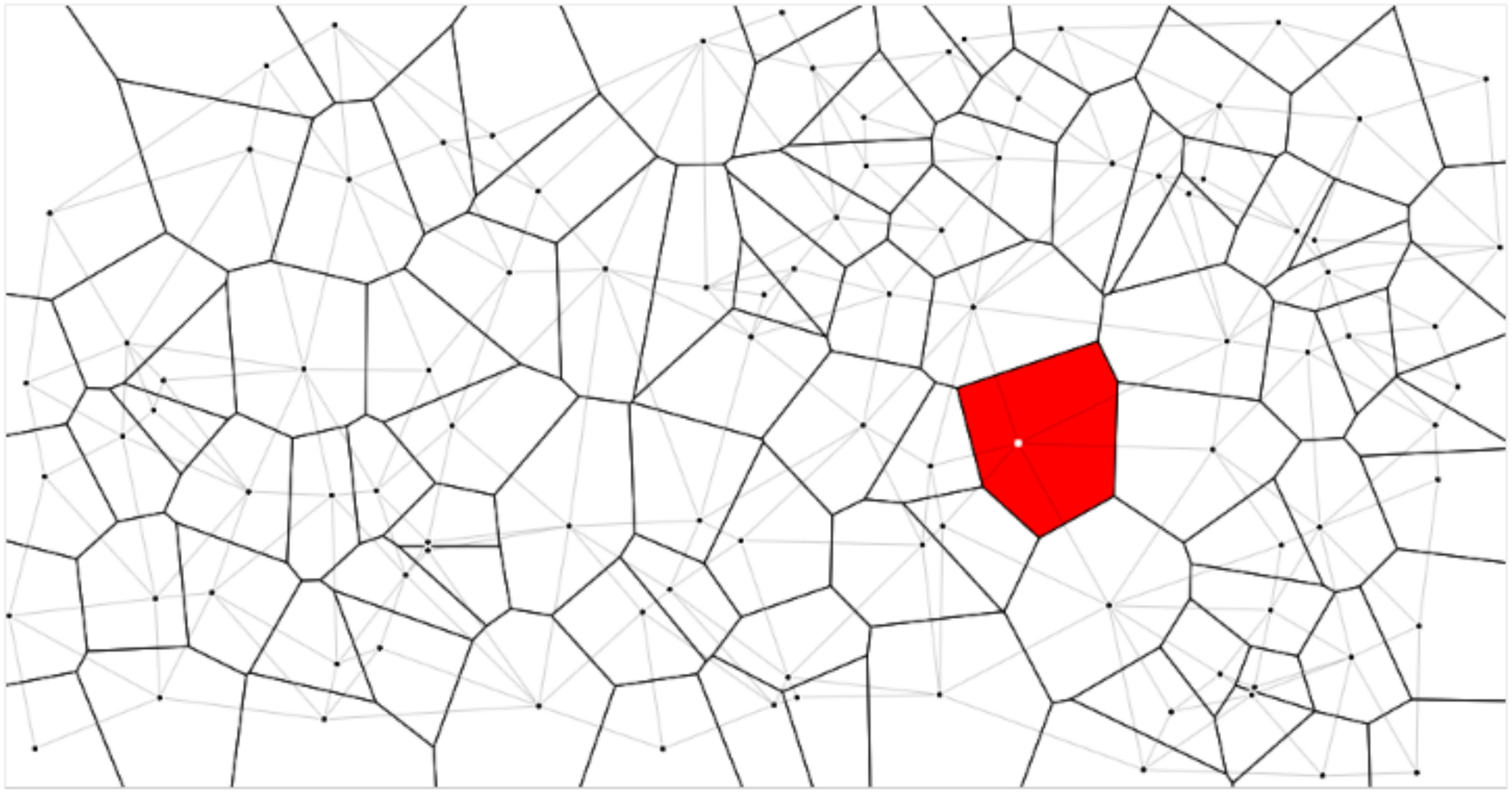
d3-chord



- Chord
- Ribbon

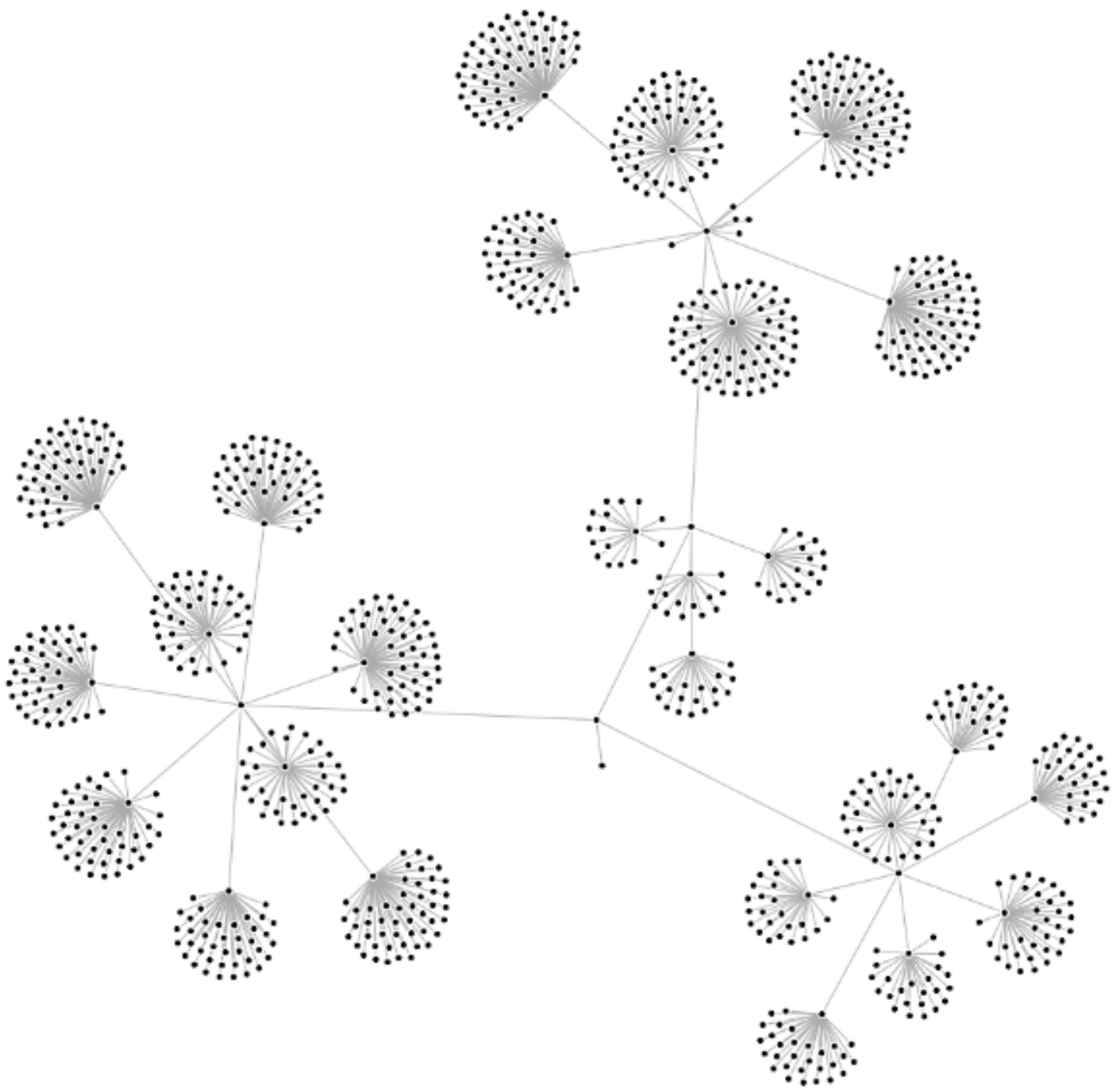
Visualize relationships or network flow with an aesthetically-pleasing circular layout.

d3-voronoi



Voronoi diagram is a partitioning of a plane into regions based on distance to points in a specific subset of the plane.

d3-force

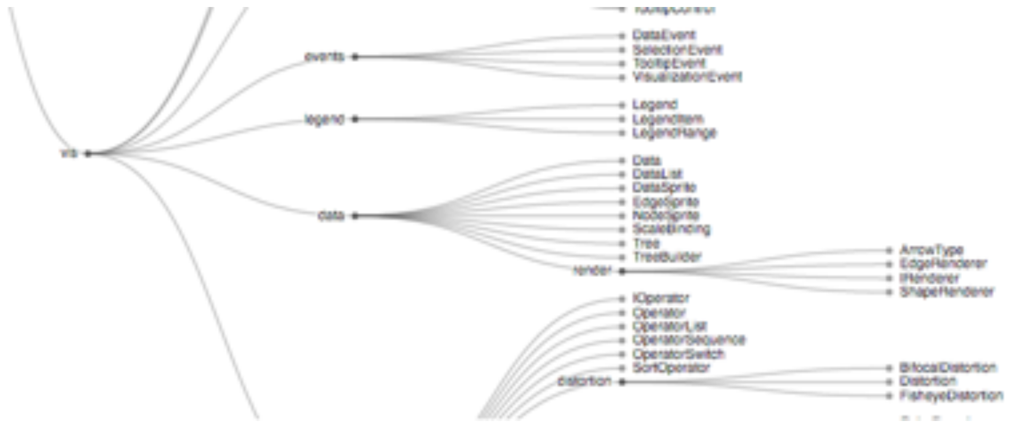


Force-directed graph.

d3-hierarchy

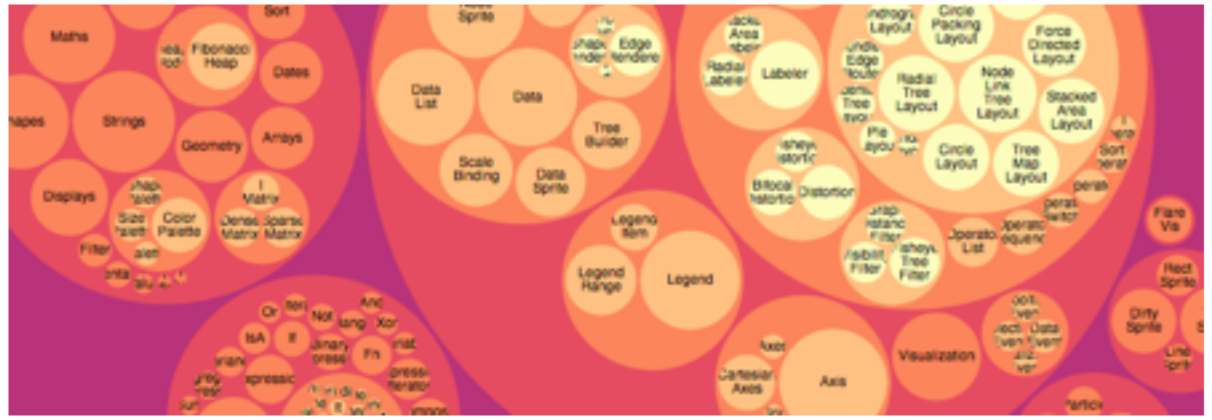
- Hierarchy (Stratify)
- Cluster - dendrograms
- Tree
- Treemap (Treemap Tiling)
- Partition
- Pack

Tree

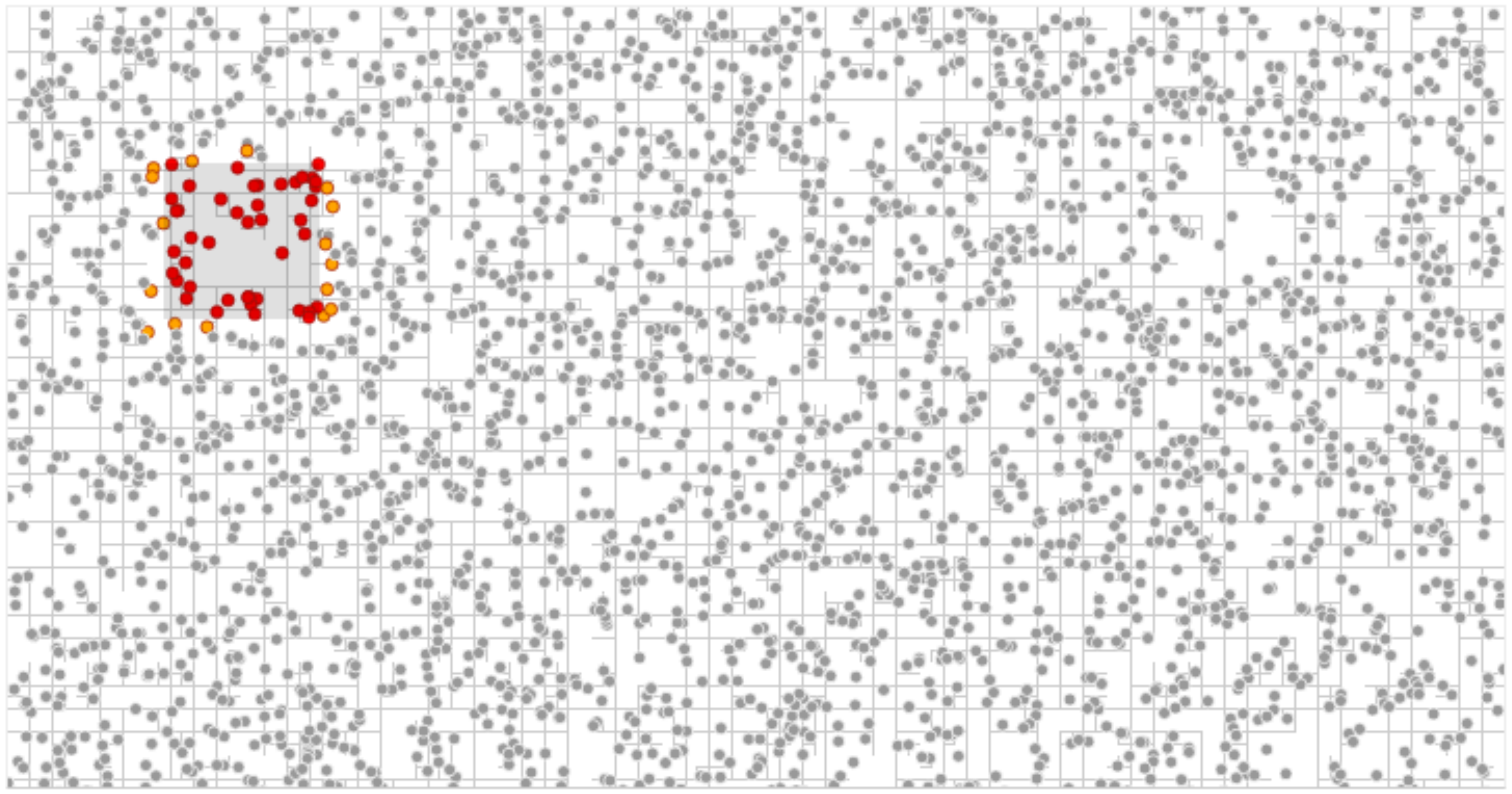


Treemap

Pack



d3-quadtree



4

- D3 now is modular
- Rewritten almost all D3 with ES2015 modules
- Flat namespace
- Renamed methods
- Many new APIs
- Examples, posts, Stack Overflow - all obsolete now

Best practices

to become D3 ninja!

- D3 is hard to learn, it doesn’t mean you can go shortcuts
- Avoid Stack Overflow, treat blog posts as feature demos, don’t copy-paste
- RTFM!
- Master SVG and CSS3
- Be aware of SVG limits, don’t put too much of elements when doing complex transformations
- Separate styles from D3 code - use classes!
    - in many examples styles are part of JS

SVG vs Canvas

- Canvas - need to redraw whole. Less smooth but better for performance at high volume elements. JavaScript based.

- SVG - gives reference to each element, animation is more smooth. Performance poor for many elements. DOM.