Introducción a visualización de datos con D3.js

Contenidos

1.

Visualización de datos

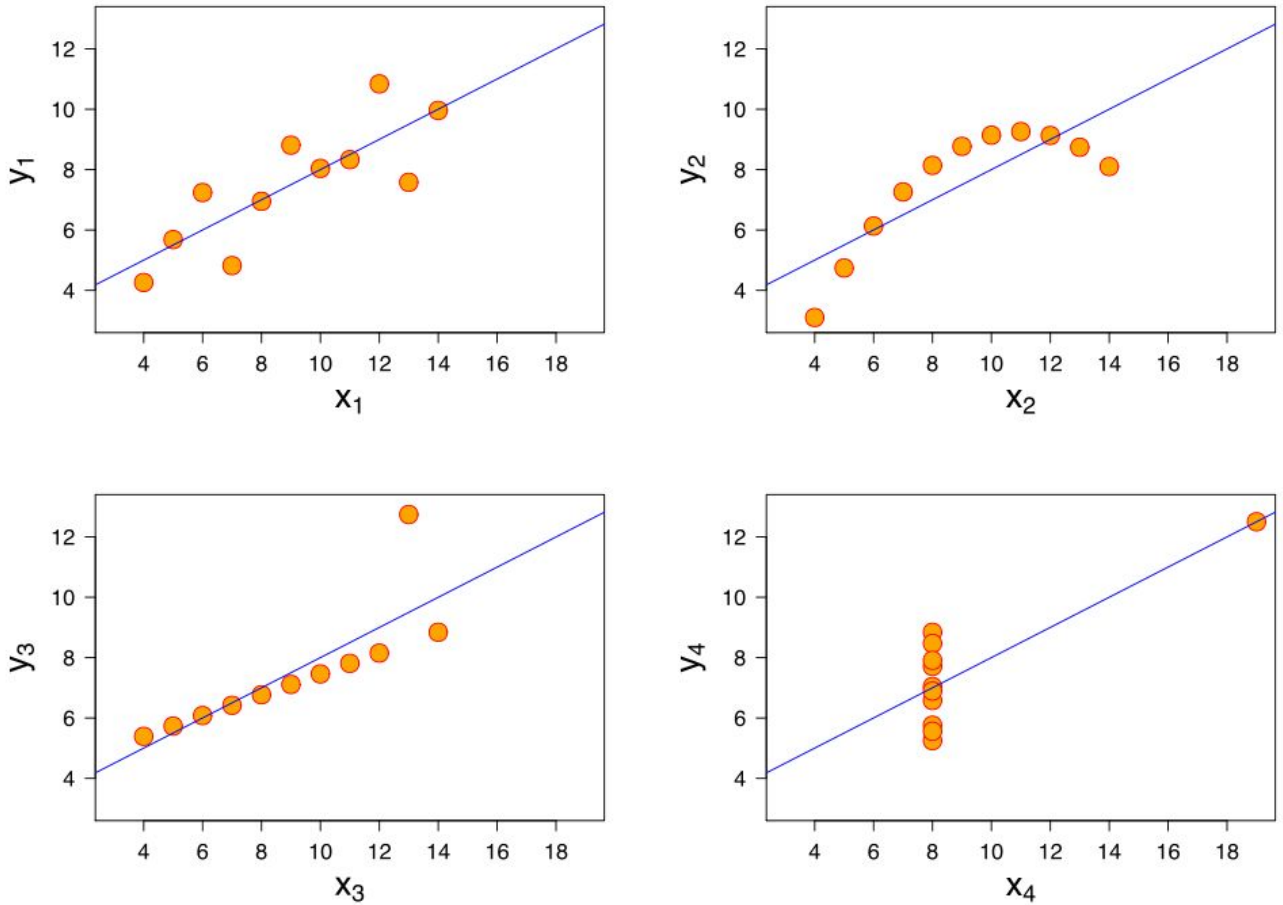
Cuarteto de Anscombe

Cuarteto de Anscombe

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Estadísticos

	Valor
Media de x	9
Varianza de x	11
Correlación	0.816
Regresión lineal	y = 3 +0.500x



https://en.wikipedia.org/wiki/Anscombe%27s_quartet#/media/File:Anscombe%27s_quartet_3.svg

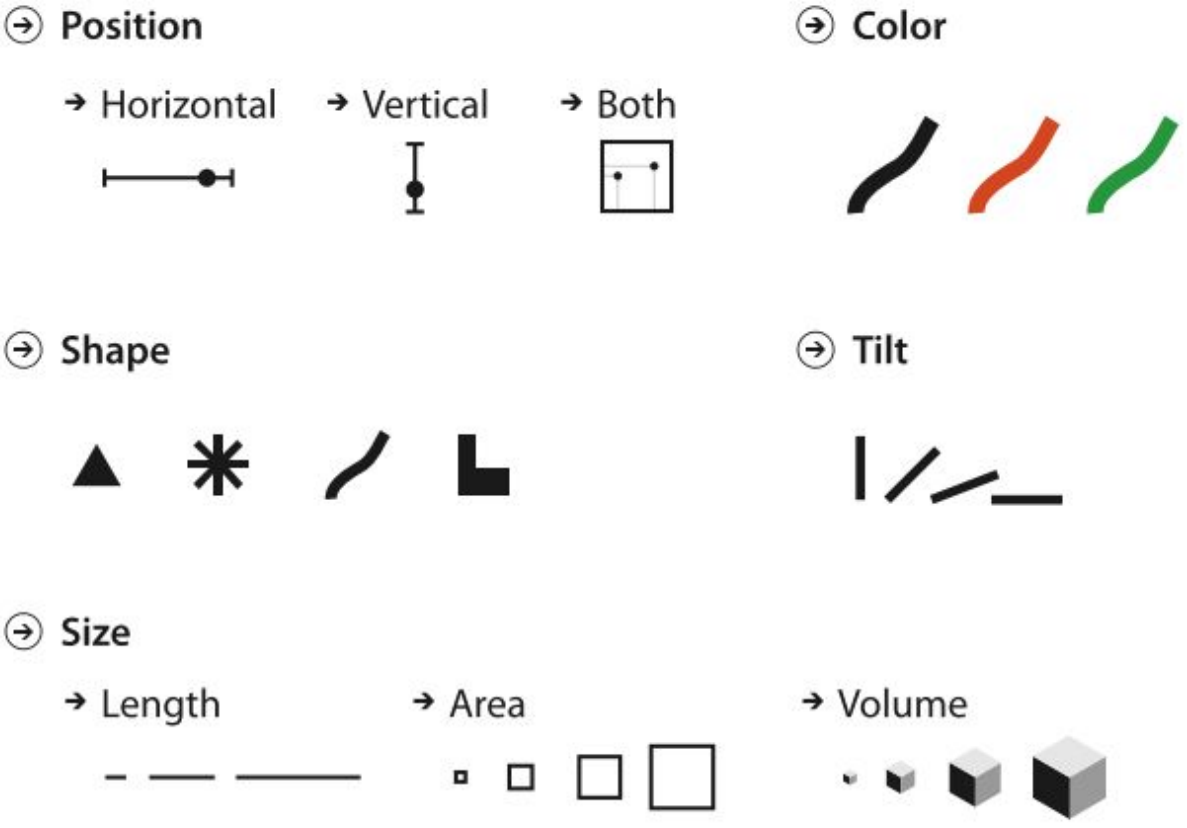
Marcas y canales

Una **marca** es un elemento gráfico básico de una imagen. Son objetos geométricos primitivos clasificables por el número de dimensiones espaciales que requieren.

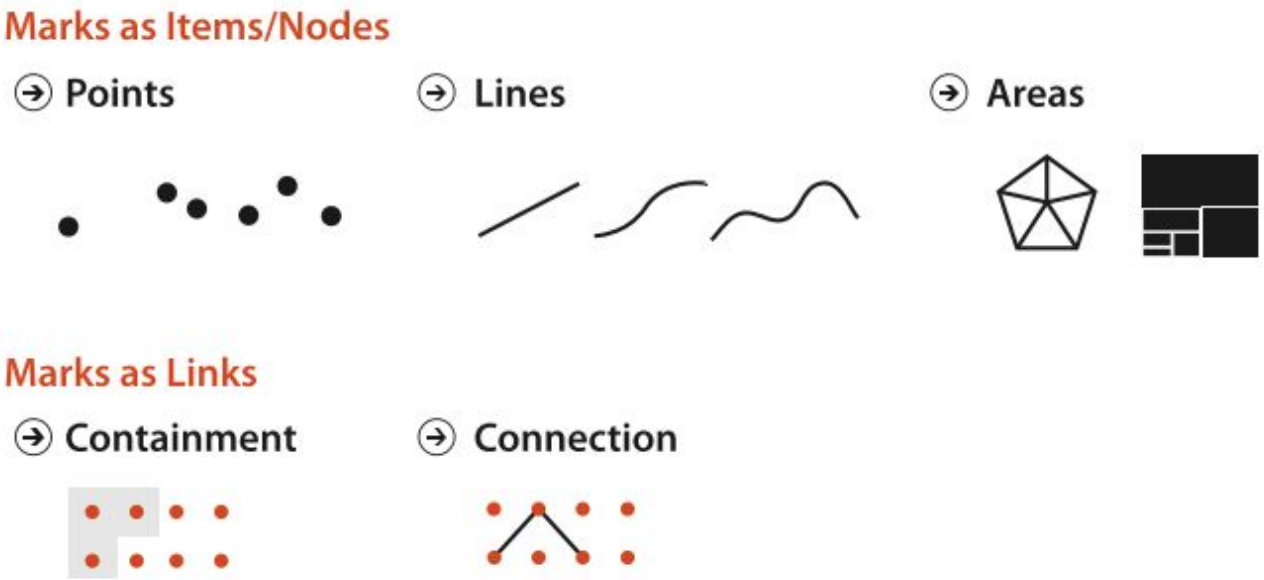
Marcas y canales

Un **canal** visual es una forma de controlar la apariencia de una marca. Existen de distintas formas, algunos corresponden a propiedades espaciales, forma, color, tamaño.

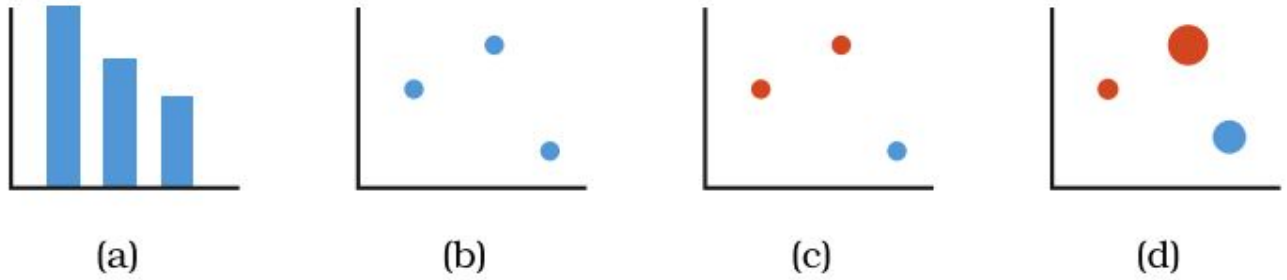
Canales


Position
Horizontal
Vertical
Both
Color
Shape
Tilt
Size
Length
Area
Volume

Marcas como ítems o nodos



Uso de marcas y canales. a) gráficos de barra codifican dos atributos con una marca de línea. b) gráficos de dispersión codifican dos atributos en la posición espacial horizontal y vertical. c) una tercer atributo categórico se codifica agregando color. d)



Expresividad y efectividad

El principio de **expresividad** dicta que la codificación visual debe expresar toda la información de los atributos de los datos, y solamente eso.

El principio de **efectividad** indica que los atributos más importantes deben ser codificados con los canales más efectivos.

La figura a continuación presenta el ranking de efectividad de los canales visuales ordenados según expresividad.

Magnitude Channels: Ordered Attributes

- Position on common scale
- Position on unaligned scale
- Length (1D size)
- Tilt/angle
- Area (2D size)
- Depth (3D position)
- Color luminance
- Color saturation
- Curvature
- Volume (3D size)

Same

Same

Most

Effectiveness

Least

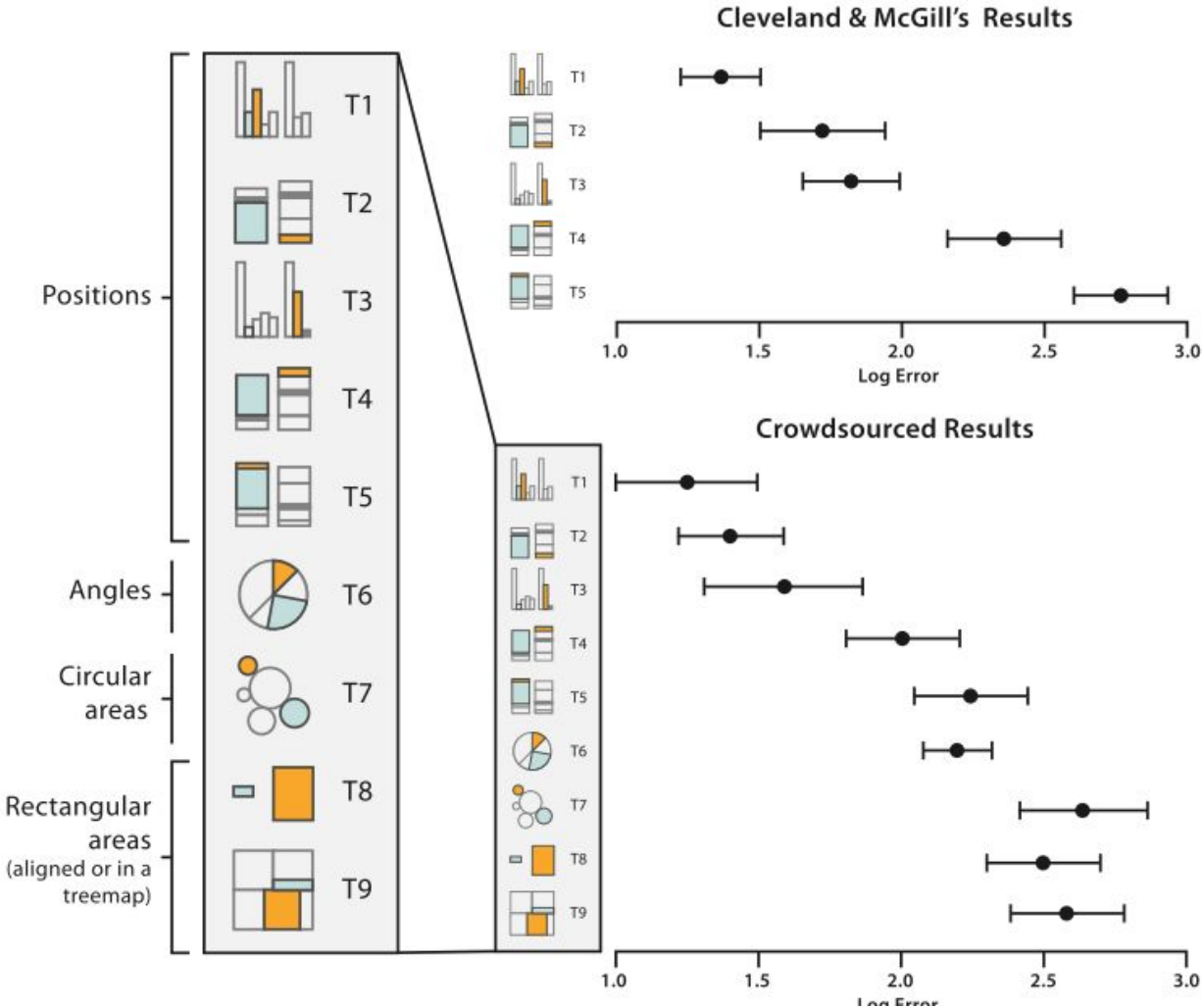
Identity Channels: Categorical Attributes

- Spatial region
- Color hue
- Motion
- Shape

Precisión de canales

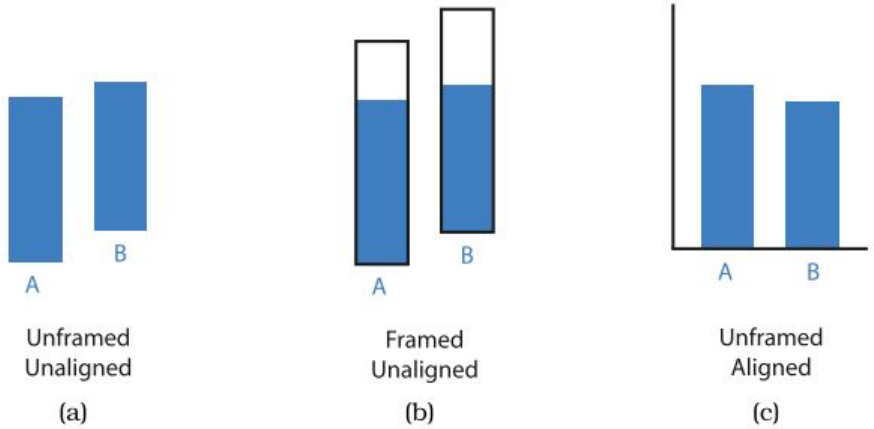
Otros experimentos respecto de la precisión de canales visuales viene de los experimentos desarrollados por Cleveland y McGill sobre los canales de magnitud.

Los resultados se muestran en la próxima imagen.


Cleveland & McGill's Results
Crowdsourced Results
Positions
Angles
Circular areas
Rectangular areas (aligned or in a treemap)
T1
T2
T3
T4
T5
T6
T7
T8
T9
1.0
1.5
2.0
2.5
3.0
Log Error

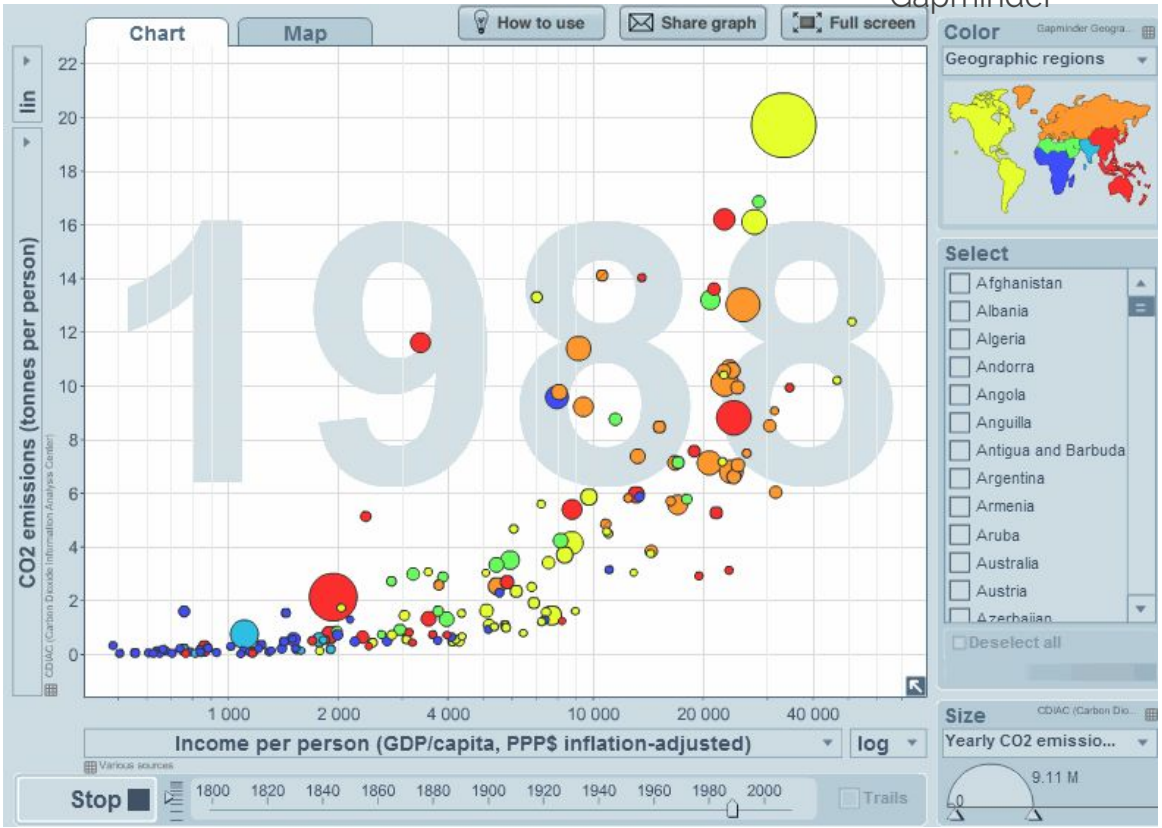
Ley de Weber

La Ley de Weber dice que juzgamos basados en diferencias relativas y no absolutas. Los largos de las barras en (a) son difícilmente comparable, agregar un marco ayuda (b), pero lo mejor es alinear las barras (c).



Hans Rosling

Gapminder



http://vialab.science.uoit.ca/dimpVis/Vis2014/Slide%20Images/Gapminder.gif

2.

Tecnologías de visualización web

Visualizaciones Web

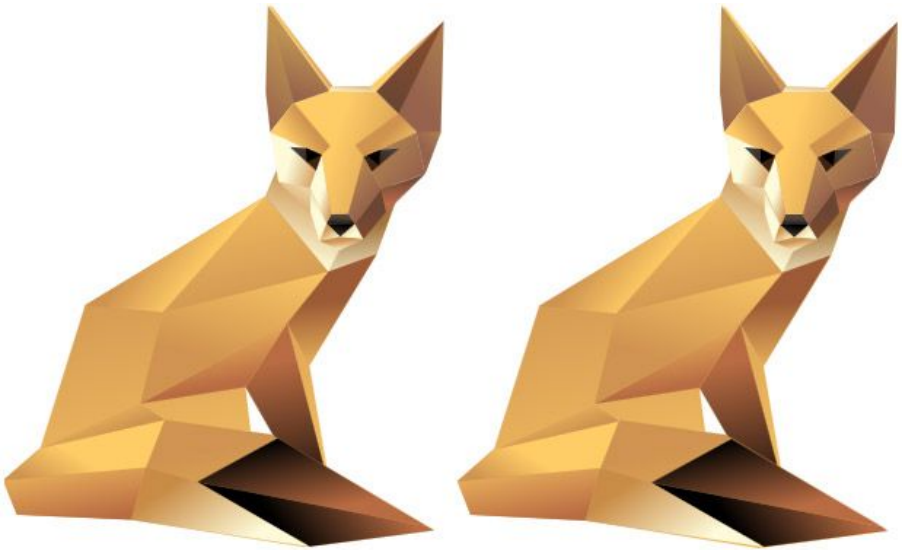
Canvas

SVG

Visualizaciones Web

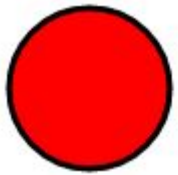
Canvas

SVG



SVG

```
<svg height="100" width="100">
          <circle cx="50" cy="50" r="40"
        stroke="black" stroke-width="3" fill="red" />
</svg>
```



Librerías para visualizaciones Javascript

Raphael.js

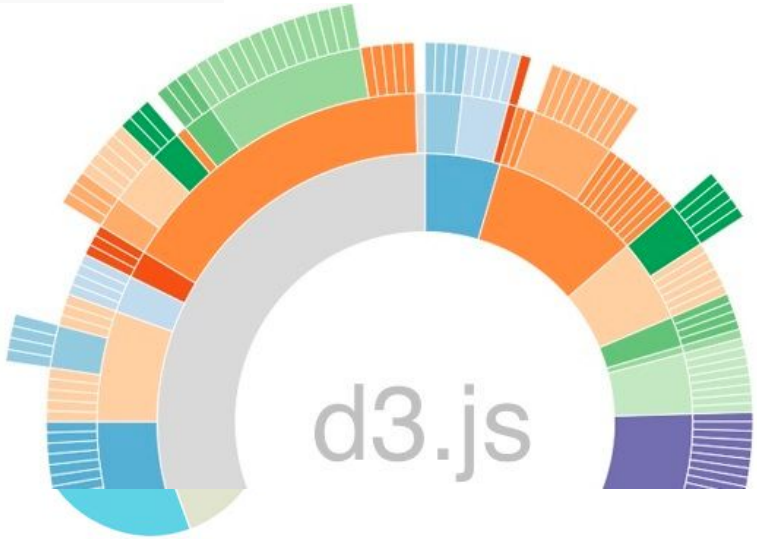


Chart.js

Easy, object oriented client side graph for designers and developers

3.

D3.js

D3.js

Librería JavaScript para visualizaciones desarrollada por Mike Bostock a partir del proyecto **Protovis**.

Permite mayor control sobre el resultado final.

Selectores

```
d3.selectAll("p")
.style("color",
"lavender")
.attr("class",
"squares")
.attr("x", 50);
```

Selectores funcionan con sintaxis CSS, similares a los de JQuery

Agregar elementos al DOM

```
d3.select("body")
.append("svg")
    .attr("width", 50)
    .attr("height", 50)
.append("circle")
    .attr("cx", 25)
    .attr("cy", 25)
    .attr("r", 25)
    .style("fill", "purple");
```

Data Binding

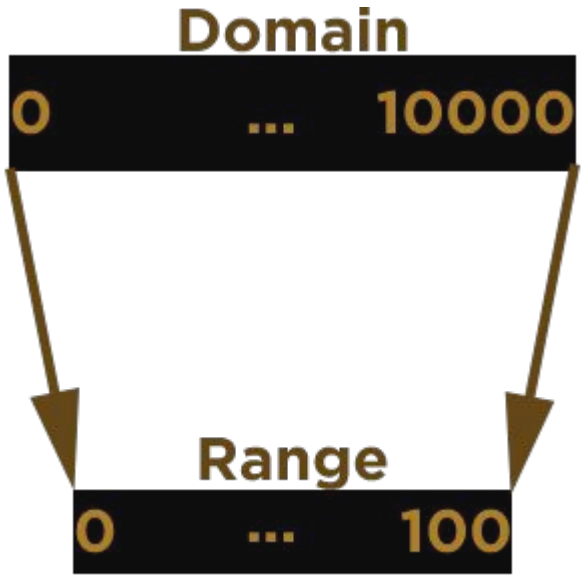
```
// Data
  var data = [
     { name:"Ireland",  income:53000, life: 78,
pop:6378, color: "green"},
     { name:"Norway",   income:73000, life: 87,
pop:5084, color: "blue" },
     { name:"Tanzania", income:27000, life: 50,
pop:3407, color: "grey" }
  ];

// Create SVG container
  var svg = d3.select("#hook").append("svg")
        .attr("width", 120).attr("height", 120)
        .style("fill", "#D0D0D0");
```

Data Binding

```
// Create SVG elements from data
svg.selectAll("circle")
      .data(data)
.enter()
.append("circle")
      .attr("id", function(d) { return d.name })
      .attr("cx", function(d) { return
      d.income/1000  })
      .attr("cy", function(d) { return d.life })
      .attr("r",  function(d) { return d.pop /1000
      *2 })
       .attr("fill",function(d){ return d.color });
```

Escalas



Si nuestros datos varían entre 0 y 10.000 y queremos que vayan entre 0 y 100, usamos una escala.

Escalas

```
var linearScale =
d3.scale.linear()
.domain([0,10000])
.range([0,100]);

linearScale(1);
//0.01
linearScale(100);
// 1
linearScale(10000)
;
// 100
```

Tipos de escalas:

1. Identidad
2. Lineal
3. Exponencial y Logarítmica
4. Cuantización y Cuantil
5. Ordinales

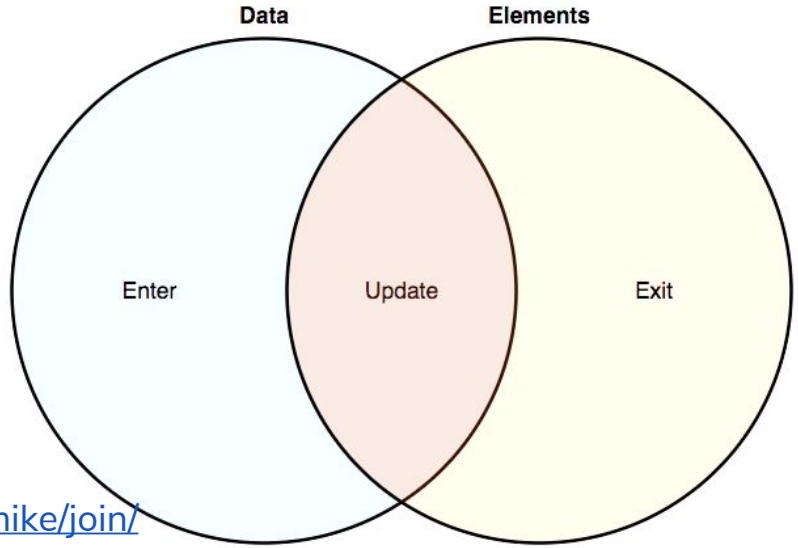
Selecciones virtuales

El operador de datos de D3.js retorna tres selectores virtuales: enter, update y exit.

El selector **enter** contiene marcadores de posición para elementos nuevos.

El selector **update** contiene a los elementos existentes.

Los elementos restantes quedan en la selección **exit** para ser eliminados.



https://bost.ocks.org/mike/join/

Gracias por su atención

Consultas a:

- diego.avila@usach.cl