

Tutorial D3.js



Data-Driven Documents

10 – Barras Apiladas y
Ordenar datos



Stacks (Apilados)

d3.stack
stack.keys

Stacks:

<https://github.com/d3/d3-shape/blob/master/README.md#stacks>



Orden datos

```
data.sort(function compare(a, b) {  
    return a - b });
```

[1,5,4] (a – b) (Ascendente)

compare(1,5); Devuelve -4 Orden a b

compare(5,4); Devuelve 1 Orden b a

[1,5,4] (b – a) (Descendente)

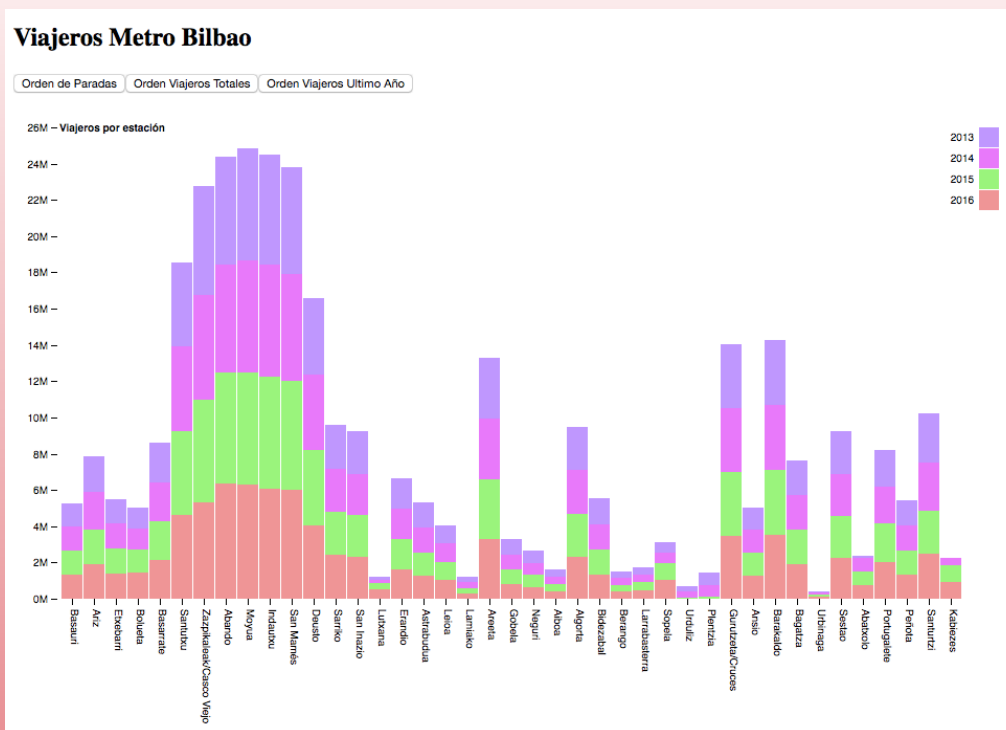
compare(1,5); Devuelve 4 Orden b a

compare(5,4); Devuelve -1 Orden a b

Sort:

https://github.com/d3/d3-selection/blob/master/README.md#selection_sort

Barras Apiladas y Botones para ordenar (I)





Barras Apiladas y Botones para ordenar (II)

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>10 Viajeros Metro Bilbao</title>
6
7   <style>
8     /* Se definen los estilos y tamaños sobre todo de los divs */
9     #graficobarras {
10       position: absolute;
11       top: 100px;
12       left: 10px;
13       width: 980px;
14       height: 610px;
15     }
16
17     .axis .domain {
18       display: none;
19     }
20   </style>
21
22   <script type="text/javascript" src="http://d3js.org/d3.v4.min.js"></script>
23
24
25   <script>
26     // Se define la función para dibujar el gráfico de barras
27     // Descargar los datos desde el csv
28     function graficobarras(){
29       d3.csv("https://raw.githubusercontent.com/irekalde/youtube/master/D3/MetroBilbao_viajeros.csv", function (d, i, columns) {
30         // += "Suma a t el valor d[columns[j]]"
31         // Agregamos una columna total y otra del orden de las paradas
32         for (j = 1, t = 0; j < columns.length; ++j) t += d[columns[j]] = +d[columns[j]];
33         d.total = t;
34         d.ordenparada = i + 1;
35         return d;
36       },
37       // La función de dibujar el gráfico.
38       function (datos) {
39         dibujar(datos);
40       });
41     };
42   </script>
```



Barras Apiladas y Botones para ordenar (III)

```
41  },
42
43  function dibujar(datos){
44    // Definir el tamaño del area dentro del svg y el grupo.
45    // Ademas se trasladar el eje de cordenadas los margenes.
46    var svg = d3.select("#graficobarras")
47      .append("svg:svg")
48      .attr("width", 960)
49      .attr("height", 590) // Porque aquí no lleva una coma?
50    margin = {top: 20, right: 20, bottom: 120, left: 40},
51    width = +svg.attr("width") - margin.left - margin.right,
52    height = +svg.attr("height") - margin.top - margin.bottom,
53    g = svg.append("g").attr("transform", "translate(" + margin.left + "," + margin.top + ")");
54
55    // Definir el eje x como una escala por bandas y distribuirla
56    var x = d3.scaleBand()
57      .rangeRound([0, width])
58      .paddingInner(0.05)
59      .align(0.1);
60
61    // Definir el eje y como una escala lineal y distribuirla
62    var y = d3.scaleLinear()
63      .rangeRound([height, 0]);
64
65    // Definir el eje y como una escala ordinal se definen una serie de colores
66    var z = d3.scaleOrdinal()
67      .range(["#ff9999", "#9fff80", "#ff80ff", "#c299ff"]);
68
69    // Con la función js slice, sacamos las claves que son los años que están en la cabecera menos la parada.
70    var claves = datos.columns.slice(1);
71
72    // Usamos la función map para que devuelva las Paradas en x
73    x.domain(datos.map(function(d) { return d.Parada; }));
74    // Aquí usamos el valor máximo del total para el y
75    y.domain([0, d3.max(datos, function(d) { return d.total; })]).nice();
76    // Usamos la escala z donde para clave año hay un color.
77    z.domain(claves);
78  }
```



Barras Apiladas y Botones para ordenar (IV)

```
79 // Generamos el grupo g para las barras apiladas
80 g.append("g")
81 .selectAll("g")
82 // Generamos el stack a partir de los datos y diciendole las keys
83 // Genera un array con un elemento por serie que tiene a su vez elementos
84 .data(d3.stack().keys(claves)(datos))
85 // Se genera un grupo por cada uno de los años y se le asigna el color correspondiente
86 .enter().append("g")
87 .attr("fill", function(d) { return z(d.key); })
88 .attr("class", "año")
89 // Y se generan los rectángulos
90 .selectAll("rect")
91 .data(function(d) { return d; })
92 .enter().append("rect")
93 .attr("x", function(d) { return x(d.data.Parada); })
94 .attr("y", function(d) { return y(d[1]); })
95 .attr("height", function(d) { return y(d[0]) - y(d[1]); })
96 .attr("width", x.bandwidth());
97 // Se genera un grupo para el eje x
98 g.append("g")
99 .attr("class", "axis")
100 .attr("transform", "translate(0," + height + ")")
101 .call(d3.axisBottom(x))
102 .selectAll("text")
103 .style("text-anchor", "start")
104 .attr("dx", "1em")
105 .attr("dy", "-0.7em")
106 .attr("transform", "rotate(90)");
107 //
108 g.append("g")
109 .attr("class", "axis")
110 .call(d3.axisLeft(y).ticks(null, "s"))
111 .append("text")
112 .attr("x", 2)
113 .attr("y", y(y.ticks().pop()) + 0.5)
114 .attr("dy", "0.32em")
115 .attr("fill", "#000")
116 .attr("font-weight", "bold")
117 .attr("text-anchor", "start")
118 .text("Viajeros por estación");
```

```
119 // Se genera la leyenda
120 var legend = g.append("g")
121 .attr("font-family", "sans-serif")
122 .attr("font-size", 10)
123 .attr("text-anchor", "end")
124 .selectAll("g")
125 .data(claves.slice().reverse())
126 .enter().append("g")
127 .attr("transform", function(d, i) { return "translate(0," + i * 20 + ")"; });
128
129 legend.append("rect")
130 .attr("x", width - 19)
131 .attr("width", 19)
132 .attr("height", 19)
133 .attr("fill", z);
134
135 legend.append("text")
136 .attr("x", width - 24)
137 .attr("y", 9.5)
138 .attr("dy", "0.32em")
139 .text(function(d) { return d; });
140
141 // Botones
142 var botones = d3.select("div#botones");
143 botones.append("button")
144 .attr("class", "boton")
145 .on("click", function() { sort("porParada"); })
146 .text("Orden de Paradas");
147
148 botones.append("button")
149 .attr("class", "boton")
150 .on("click", function() { sort("porTotal"); })
151 .text("Orden Viajeros Totales");
152
153 botones.append("button")
154 .attr("class", "boton")
155 .on("click", function() { sort("porUltimo"); })
156 .text("Orden Viajeros Ultimo Año");
157
158
```



Barras Apiladas y Botones para ordenar (V)

```
159 // Funciones Ordenar
160 function sort(orden){
161     if (orden == "porParada"){
162         datos = datos.sort(function (a, b) {
163             return a.ordenparada - b.ordenparada;
164         });
165     }
166     else if (orden == "porTotal"){
167         datos = datos.sort(function (a, b) {
168             return b.total - a.total;
169         });
170     }
171     else {
172         datos = datos.sort(function (a, b) {
173             return b[2016] - a[2016];
174         });
175     }
176 };
177
178 d3.select("svg").remove();
179 d3.selectAll("button").remove();
180 dibujar(datos);
181
182 };
183
184
185
186 };
187
188
189 </script>
190 </head>
191
192 <body onload="graficobarras()">
193 <!-- Se generan los divs para alojar los gráficos -->
194 <div id="titulo"><H2>Viajeros Metro Bilbao</H2></div>
195 <div id="botones"></div>
196 <div id="graficobarras"></div>
197 </body>
198 </html>
```