

Más allá del core

- ▶ **Son funciones que nos facilitan el mapeo de datos a variables visuales**
- ▶ En el API se llaman scales
- ▶ Se utilizan sobre todo para relacionar el **dominio** de los datos con el **rango** de valores que tomarán al mapearlos a píxeles

```

var w = 500;
var h = 2000;
var barMargin = 1;

var svg = d3.select(".chart")
  .append("svg")
  .attr("width", w)
  .attr("height", h);

d3.csv("cars.csv", function (cars) {
  render(cars);
});

var render = function(datos) {

  var weights = datos.map(function(d){
    return parseInt(d['weight (lb)']);});

  /**
   * Adaptado al alto del SVG
   *
   * El redondeo evita problemas de aliasing
   */
  var barWidth = Math.floor( (h / weights.length) - barMargin );

  svg.selectAll('.bar')
    .data(weights)
    .enter()
    .append('rect')
    .attr('class', 'bar')
    .attr('y', function(d,i) {return i*(barWidth+barMargin);})
    .attr('width', function(d){return d * 0.05;})
    .attr('height', barWidth);
};

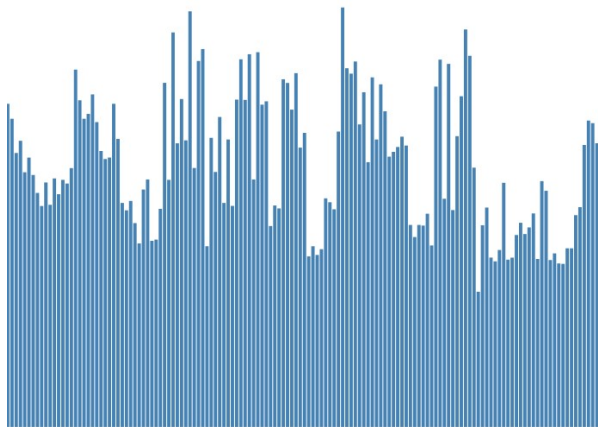
```

Ejercicio 11

- ▶ Sobre el código del Ejercicio 10:
 - ▶ Ajustar los tamaños, horizontales y verticales, de las barras utilizando una escala lineal

Ejercicio 12

- Orienta el diagrama de barras para que las barras queden verticales



Layouts

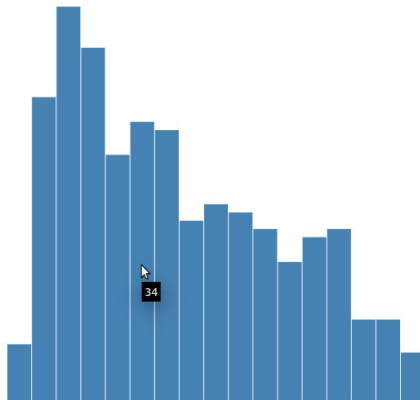
- ▶ Los layout no son más que una transformación de datos, **no dibujan nada**.
- ▶ Ayudan a calcular los valores de las variables gráficas que van a representar los datos

Tooltips

- ▶ Son pequeños textos que nos dan información de detalle sobre un elemento.
- ▶ Se muestran cuando nos posamos sobre algún elemento.
- ▶ La manera más fácil es utilizar el elemento **title** de SVG

```
<circle><title>Este el mensaje del tooltip</title></circle>
```

Ejercicio 13



- ▶ Utiliza el **layout histogram** para crear un histograma de pesos con 20 barras
- ▶ Para definir los 20 bins utiliza el método **ticks** de tu escala horizontal
- ▶ Añade un tooltip en cada barra con su valor (el peso del coche)

Ejes

- ▶ Son funciones del módulo **d3.svg** que **pintan** los ejes
- ▶ En el API se llaman axes
- ▶ Se utilizan siempre en conjunción con las **escalas**
- ▶ Desde la selección del elemento padre ejecutamos nuestro eje ya configurado.

```
var yAxis = d3.svg.axis() // Es un clousure
    .scale(y)
    .orient("left");

svg.call(yAxis);
```

- Es necesario darles un estilo con CSS:

```
.axis text {  
  font: 10px sans-serif;  
}  
  
.axis path,  
.axis line {  
  fill: none;  
  stroke: #000;  
  shape-rendering: crispEdges;  
}
```

Groups en SVG

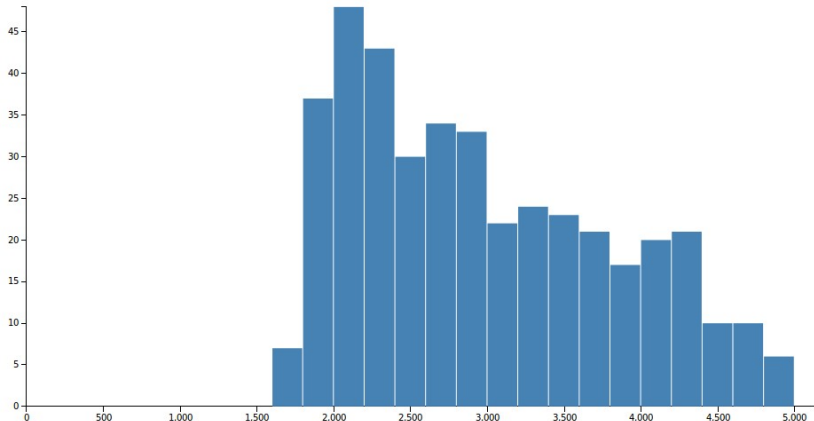
- ▶ El elemento **g** es sólo un contenedor
- ▶ Se puede utilizar para hacer **capas** (SVG se renderiza con el algoritmo del pintor)
- ▶ Muy utilizado para aplicar transformaciones a muchos elementos

Márgenes

Existe una convención para poner márgenes a la gráfica. Útil para poner los ejes en ellos.

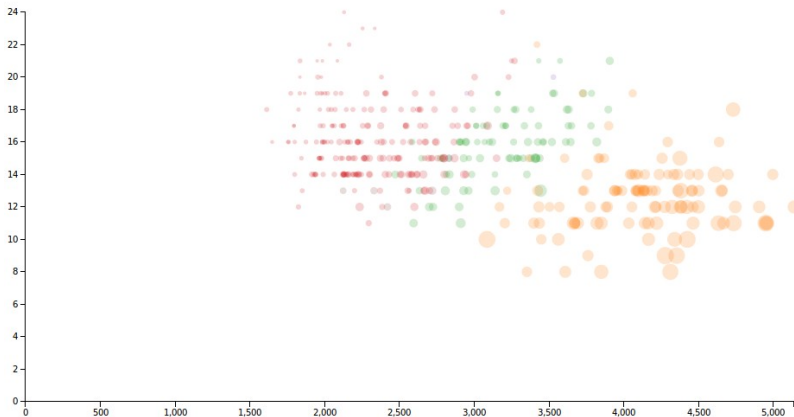
```
var margin = {top: 10, right: 30, bottom: 30, left: 30};  
var width = 960 - margin.left - margin.right;  
var height = 500 - margin.top - margin.bottom;  
  
var svg = d3.select("body").append("svg")  
  .attr("width", width + margin.left + margin.right)  
  .attr("height", height + margin.top + margin.bottom)  
  .append("g")  
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
```

Ejercicio 14



- ▶ Añade dos ejes con **svg.axis**, a la coordenada X y a la Y
- ▶ Utiliza la convención de márgenes para situar los ejes fuera de la gráfica
- ▶ Mejora la legibilidad de los ejes con CSS

Ejercicio 15



Interacción

- ▶ Cualquier elemento del DOM puede ser interactivo
- ▶ Con d3, cualquier selección tiene un método **on** para capturar eventos y tratarlos con callbacks

```
d3.selectAll("rect").on("click", function(d){});
```

- ▶ Lista de eventos comunes en SVG: click, mousedown, mouseup, mouseover, mouseout
- ▶ Funciones útiles: **d3.mouse** y **d3.event**

Ejercicio 16

- ▶ Utiliza la metaclassa “:hover” para dibujar un stroke

```
p:hover { color:red; } /* Código CSS */
```

- ▶ Cuando se pincha (on(“click”, callback)) en un coche se muestra un mensaje con el mandato “alert” que tiene el número de cilindros del coche

```
// Prueba en la consola de javascript  
alert("hola mundo");
```

- ▶ Cuando se pasa por encima con el ratón, peso y velocidad se muestran en un “p” arriba de la gráfica
 - ▶ Añade el p en el html con una clase para identificarlo
 - ▶ El evento a capturar es **mouseover**
 - ▶ Modifica el **text** del **p** desde d3