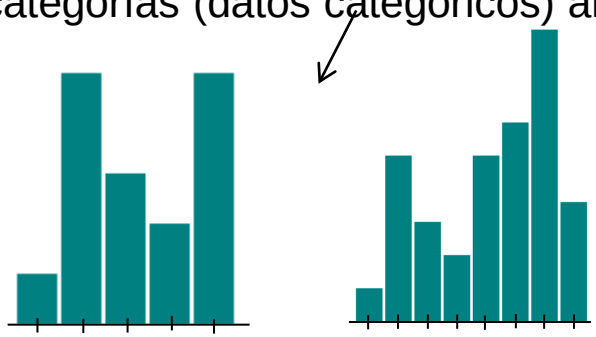
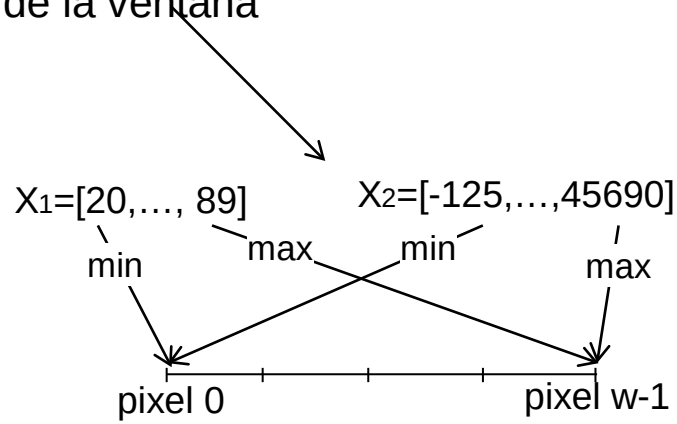
Índice

Ajustando: d3-scale. Introducción

- El problema: ajustar la visualización al tamaño prefijado de ventana.
  - Eje X: hay que ajustar los valores (datos numéricos) o el número de categorías (datos categóricos) al ancho de la ventana

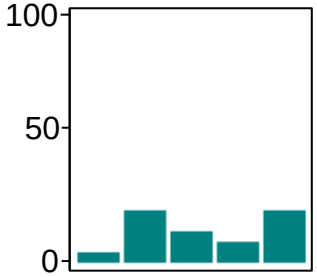


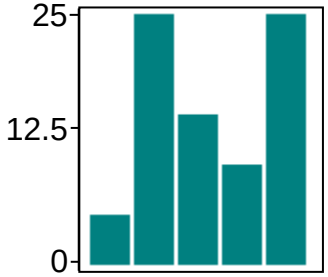
Cálculo posiciones.
Ancho=W, número barras=N
división en N bandas: p = (W-1) / N
posiciones barras, centro bandas: p/2, p+(p/2), 2p+(p/2), …



  - Eje Y: ajustar altura de la representación para que ocupen el espacio disponible y no quede desproporcionada
    - Ej. Gráfico de barras. Datos=[5, 25, 15, 10, 25]. Alto ventana: 100 px

Se dejan valores originales:
Mal ajuste.





Se ajusta altura:
c = Alto ventana / max(Datos) = 4
Se multiplica cada dato por *c (4)*

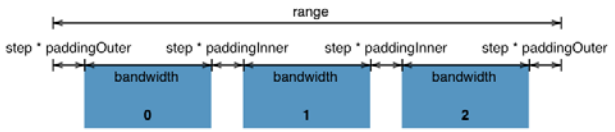
Ajustando: d3-scale. Conceptos Generales

- D3-scale: generador de funciones que permite realizar el ajuste de los datos a un rango y escala predeterminados.
- API: https://github.com/d3/d3/blob/master/API.md#scales-d3-scale
- Conceptos importantes:
  - Dominio de entrada (*domain*).
    - Normalmente son los datos a representar en el eje X o en el eje Y.
      - Pero no tiene por qué, ya que se puede aplicar escalas a cualquier transformación de datos que queramos hacer
    - Puede ser continuo ([minEnt,maxEnt]) o discreto
  - Rango de salida (*range*). También puede ser continuo o discreto
    - Continuo ([minSal,maxSal]).
      - Normalmente es el tamaño en pantalla de la zona del gráfico.
    - Discreto. Su significado depende del tipo de escala. Veremos varios ejemplos
    - Funciones relacionadas:
      - *escala*.rangeRound([rango]): la salida es redondeada al entero más cercano
      - *escala*.clamp(true): para que el valor de salida no se salga del rango de salida especificado

Ajustando: d3-scale. Escalas

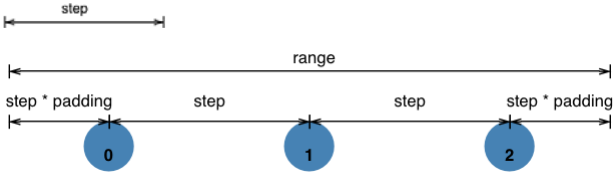
- D3 ofrece múltiples tipos de escalas, con múltiples opciones en cada una
- Algunas escalas interesantes:
  - Lineal (https://github.com/d3/d3-scale/blob/v3.2.2/README.md#scaleLinear):
    - D3.scaleLinear(([[domain,]range]))
    - Realiza una ajuste continuo y lineal del dominio de entrada al rango de salida
  - Intervalos (Quantize, https://github.com/d3/d3-scale/blob/master/README.md#scaleQuantize):
    - d3.scaleQuantize([[domain,]range])
    - Divide el Dominio de entrada continuo en tantos intervalos como elementos tenga el rango de salida discreto.
  - Logarítmica (https://github.com/d3/d3-scale/blob/v3.2.2/README.md#scaleLog):
    - d3.scaleLog([[domain,]range])
    - Crea una escala logarítmica
  - Bandas y puntos
    - Divide un rango de salida continuo en intervalos de bandas o puntos asociados a cada elemento del dominio de entrada discreto
    - d3.scaleBand([[domain,]range])

(https://github.com/d3/d3-scale/blob/master/README.md#scaleBand)



- d3.scalePoint([[domain,]range])

(https://github.com/d3/d3-scale/blob/master/README.md#scalePoint)



Ajustando: d3-scale - Ejemplos

- Aclaración del código: las distintas funciones generadoras de escalas generan una función, que pude ser almacenada en una variable. En javascript esto es posible

Escala Lineal

```
var escalaLineal = d3.scaleLinear([20, 89], [0, 350]);
console.log('Salida para 20: ' + escalaLineal(20));
console.log('Salida para 65: ' + escalaLineal(65));
// Prueba rangeRound
escalaLineal.rangeRound([0, 350]);
console.log('Salida para 65 round: ' + escalaLineal(65));
// Prueba clamp con un dato fuera de dominio
console.log('Fuera dominio: ' + escalaLineal(10));
escalaLineal.clamp(true);
console.log('Fuera dominio clamp: ' + escalaLineal(10));
```

Escala Intervalo

```
//Usamos métodos domain/range
//Para mostrar la otra opción
var escalaQuantize = d3.scaleQuantize()
    .domain([20,89])
    .range(['bajo','medio','alto']);
console.log('Categoria del 30: ' + escalaQuantize(30));
console.log('Categoria del 66: ' + escalaQuantize(66));
```

Escala de Puntos

```
var escalaPuntos = d3.scalePoint()
    .domain(['uno','dos','tres','cuatr'])
    .range([0, 350]);
console.log('Pixel punto uno: ' + escalaPuntos('uno'));
console.log('Pixel punto dos: ' + escalaPuntos('dos'));
```

Escala de Bandas

```
var escalaBandas = d3.scaleBand()
    .domain(['uno','dos','tres','cuatr'])
    .range([0, 350]);
console.log('Inicio banda uno: ' + escalaBandas('uno'));
console.log('Inicio banda dos: ' + escalaBandas('dos'));
```

Paleta

- Algunas paletas estándar de D3:
  - Documentación: https://github.com/d3/d3/blob/master/API.md#color-schemes-d3-scale-chromatic
  - Ver las escalas: https://bl.ocks.org/pstuffa/3393ff2711a53975040077b7453781a9

Ej.: d3.schemeCategory10

d3.schemeCategory20

d3.schemeCategory20b

Ejes – “Manual”

- Se pueden construir “a mano”
  - Mediante comandos SVG que construyen las líneas y las marcas de los ejes.
  - Ej. de los ejes:

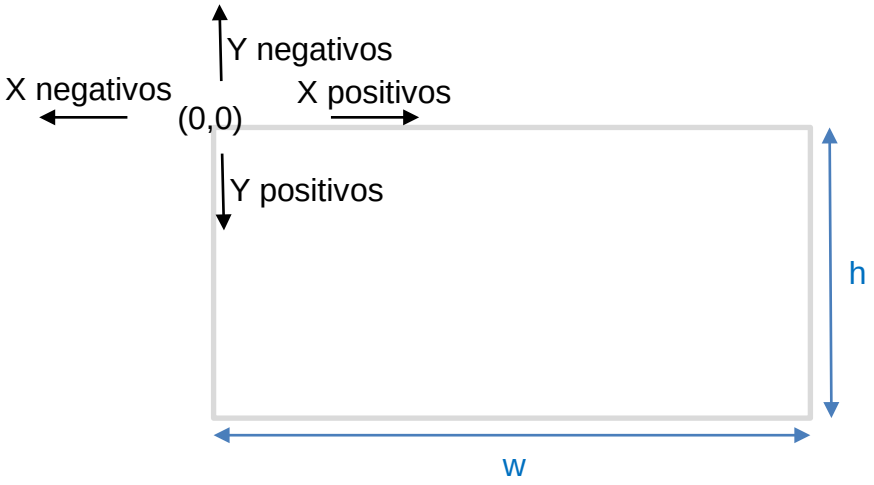
```
var datos = [5,10,15,20,25];
var svg = d3.select("body").append("svg");
var w = 400;
var h = 300;
var margin_x = 32;
var margin_y = 20;
svg.attr("width", w).attr("height", h);  //Fijo tamaño alemento SVG
var y = d3.scaleLinear().domain([0, d3.max(datos)]).range([0 + margin_y, h - margin_y]);  // Ajuste eje Y
var x = d3.scaleLinear().domain([0, datos.length]).range([0 + margin_x, w - margin_x]);  // Ajuste eje X
var g = svg.append("svg:g").attr("transform", "translate(0," + h + ")");  // Grupo svg y traslación del origen de coordenadas
g.append("svg:line").attr("x1", x(0)).attr("y1", -y(0)).attr("x2", x(datos.length)).attr("y2", -y(0)).attr("stroke","black"); // eje X
g.append("svg:line").attr("x1", x(0)).attr("y1", -y(0)).attr("x2", x(0)).attr("y2", -y(d3.max(datos))-10).attr("stroke","black"); // Eje Y
```

Explicación en siguiente transparencia

  - Faltarían las marcas (ticks), sus valores asociados y la rejilla
    - Esto se puede ver en el documento en la web: D3_Tema3_dibujandoEjes.pdf

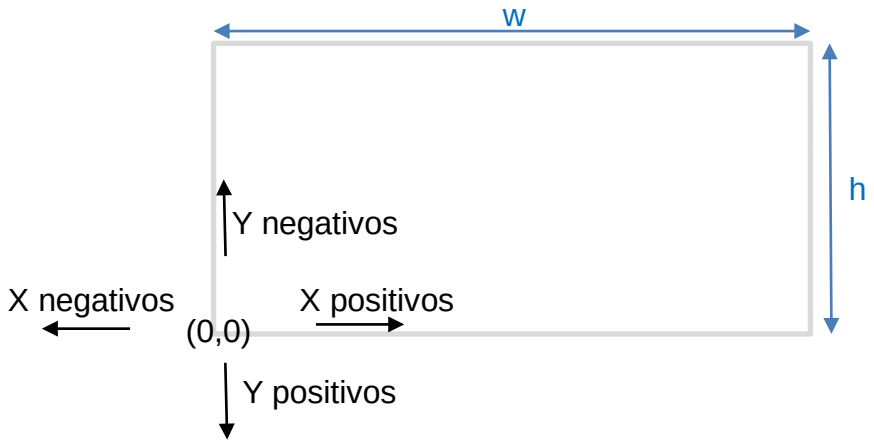
Explicación “Translate”

Coordenadas en la ventana svg original



Coordenadas del grupo en la ventana svg tras “translate”

svg.append("svg:g").attr("transform", "translate(0," + h + ")")



Ejes – "Automático" (I)

- Documentación general: https://github.com/d3/d3/blob/master/API.md#axes-d3-axis
- Generadores de ejes: d3.axisTop, d3.axisRight, d3.axisBottom, d3.axisLeft
  - Ej. construir los ejes y marcas:

```
var datos = [5,10,15,20,25];
var w = 336; // Ancho del gráfico
var h = 260; // Alto del gráfico
var margin_x = 32;
var margin_y = 20;
var x = d3.scaleLinear().domain([0, datos.length]).range([0,w]);  // Ajuste eje X
var y = d3.scaleLinear().domain([0, d3.max(datos)]).range([h,0]);  // Ajuste eje Y
var xAxis = d3.axisBottom(x).ticks(5);  // Defino la función eje X
var yAxis = d3.axisLeft(y).ticks(5);  // Defino la función eje Y
var svg = d3.select("body").append("svg").attr("width", w + margin_x + margin_x)  // Elementos svg
        .attr("height", h + margin_y + margin_y).append("g")
        .attr("transform", "translate(" + margin_x + "," + margin_y + ")");
svg.append("g").attr("class", "x axis").attr("transform", "translate(0," + h + ")").call(xAxis);  // Eje X
svg.append("g").attr("class", "y axis").call(yAxis);  // Eje Y
```

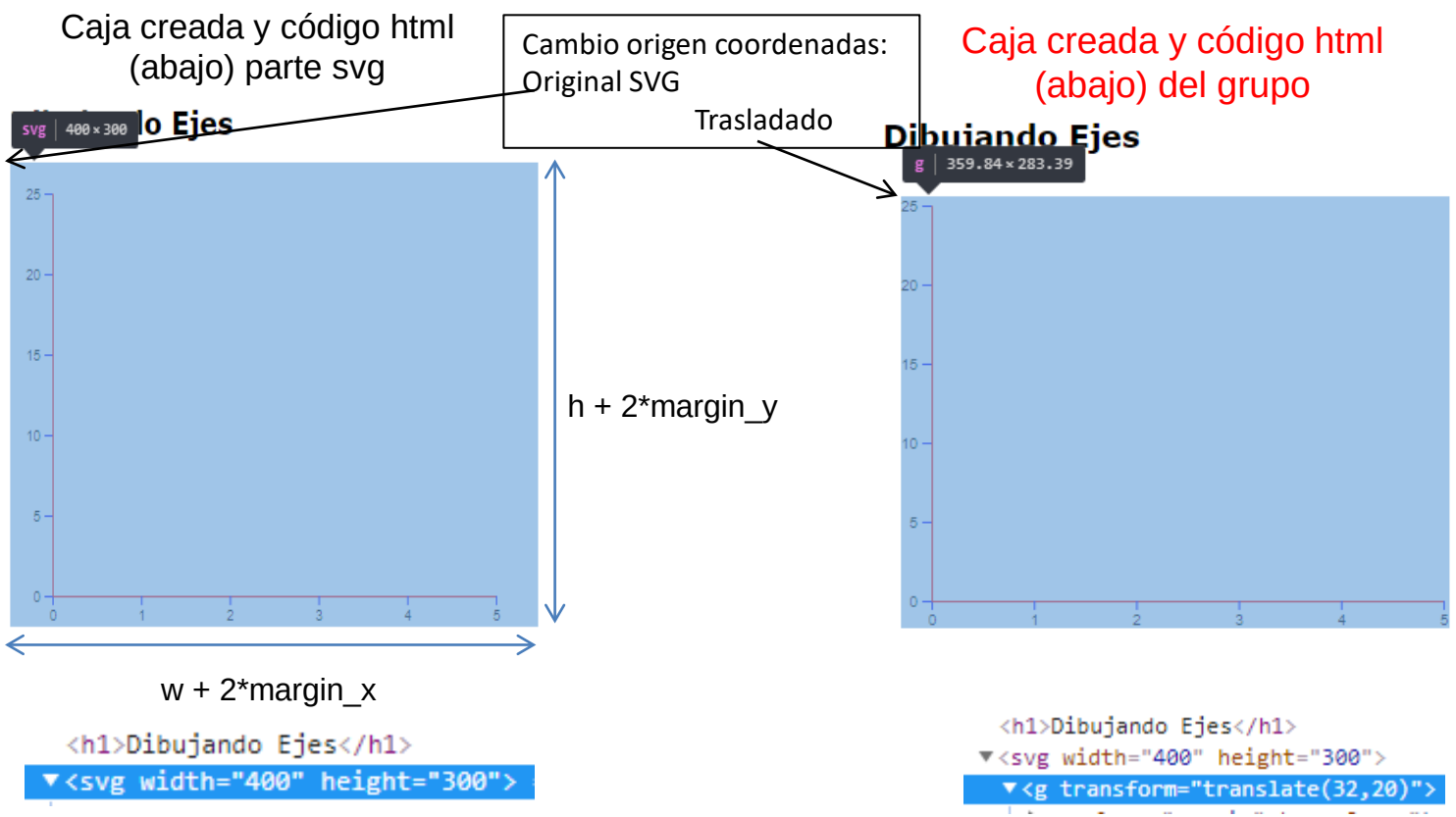
Comentario: En el eje Y se va a representar la altura de las bandas. Ver como se han cambiado los valores en el rango, para adaptarse a las coordenadas SVG (0 en la esquina superior izquierda)

  - Al ejecutar este código es interesante para entender mejor lo hecho inspeccionar mediante las herramientas de desarrollador el código HTML creado. Vamos a hacerlo en las siguientes transparencias, en las partes más importantes.

Ejes – "Automático" (II)

- Análisis código creación automática de ejes.
  - Caja contenedora svg (figura de la izquierda, código con texto en negro) y caja del gráfico, mediante grupo que traslada origen (figura derecha, código con texto rojo)

```
var svg = d3.select("body").append("svg").attr("width", w + margin_x + margin_x).attr("height", h + margin_y + margin_y)
        .append("g").attr("transform", "translate(" + margin_x + "," + margin_y + ")");
```

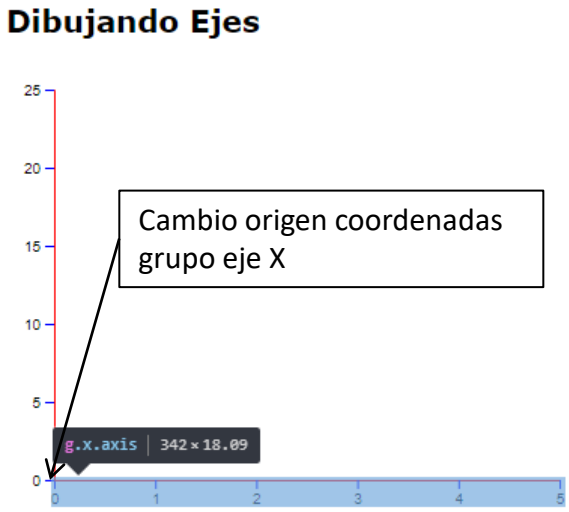


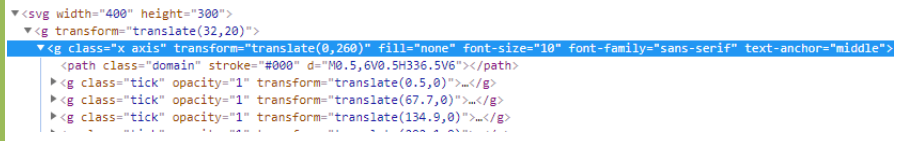
Ejes – "Automático" (III)

- Análisis código creación automática de ejes (Continuación).
  - Dibujando los ejes:

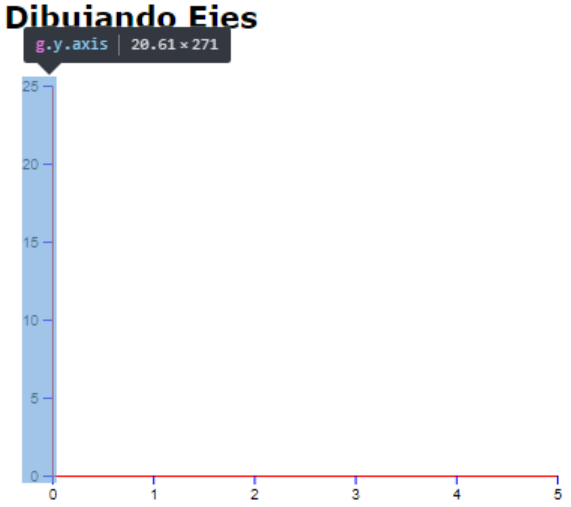
```
svg.append("g").attr("class", "x axis").attr("transform", "translate(0," + h + ")").call(xAxis);  // Eje X

svg.append("g").attr("class", "y axis").call(yAxis);  // Eje Y
```

Eje X. Caja y Código html





Eje Y. Caja y Código html



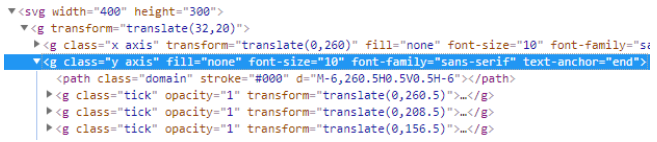


Gráfico de Barras

- Versión básica: páginas 97 a 104 del documento en la web D3_Tema3_dibujandoGraficosDeBarras.pdf
  - Ahí se puede ver, también, un ejemplo de uso de la paleta de colores.
  - Para entender mejor lo realizado es conveniente ver el código html final generado
  - El libro está en la versión 3, pero se ha añadido el código para que funcione en las versiones superiores (4, 5 y 6).