

Cápsula 3: Marcas en D3.js II

Hola, bienvenidxs a una cápsula del curso Visualización de Información. Esta corresponde a la segunda parte sobre la generación de marcas en D3.js.

Ya revisamos cómo generar marcas pequeñas con formas distintas, y cómo generar líneas a partir de datos. En esta cápsula revisaremos cómo generar áreas complejas a partir de datos.

Podemos pensar en un gráfico de línea cuya área bajo la línea está coloreada de algún color como nuestro objetivo. A eso se le conoce generalmente como gráfico de área, porque pone **énfasis en el espacio bajo la línea**. Generalmente se usa cuantificando un valor cuyos valores se suman o acumulan, y generan un gran total.

D3 nos provee también una forma de generar áreas a partir de datos para este fin, y sigue una lógica similar a la ocupada para definir líneas. Específicamente, **un generador de áreas se define a partir de dos líneas, las cuales une en sus extremos para generar una región**.

En pantalla tengo un programa que se basa en datos muy similares a los usados para definir una línea en la cápsula pasada. Pero ahora, hay una definición para un generador de áreas. A este le definimos funciones de acceso para definir las líneas que formarán el área, con los métodos "x", "y0" e "y1".

Los métodos "y0" e "y1" definirán por separado las coordenadas verticales utilizadas para las dos líneas que definen áreas. También existen los métodos "x0" y "x1" que hacen lo mismo de forma respectiva para coordenadas horizontales, pero en este caso podemos definir ambos utilizando un único método "x", que define de la misma forma las coordenadas horizontales de ambas líneas.

Este generador entonces definirá que la línea 1 seguirá los valores de los atributos de los datos usados, en base a las escalas definidas arriba, y la línea 0 mantendrá su posición vertical y en el 0 de la escala en Y. No hay un orden específico de cómo definir estas líneas, pero por lo general se usa la línea 0 para definir el cero o base de un área.

Si lo probamos, veremos que se genera un área para un gráfico de área. La línea superior corresponde a la línea 1, y la inferior a la línea 0. En este caso solo una de las líneas es curva ya que definí que los valores verticales de la línea 0 son constantes.

Pero puedo alterar perfectamente esto. Para probar definiré que "y0" será el valor del mismo atributo usado en "y1" pero restando en 0.5. Si lo probamos, obtenemos una franja curva de altura constante, ya que la diferencia de altura entre los puntos de las líneas es constante.

Bajo esa idea, esto da el potencial para definir áreas apiladas una sobre la otra. Agregaré al código una sección casi idéntica, pero que traslada aún más las líneas de una segunda área. Si lo probamos, obtenemos la segunda área de la misma forma pero bajo la primera.

Esto nos da la intuición de cómo generar un gráfico de áreas acumuladas. Un problema usual con esto es que debemos preprocesar datos, especialmente sumar valores unos detrás de otros para determinar luego las alturas correspondientes de las líneas que generan las áreas. D3 nuevamente viene al rescate ya que provee una función para hacer estos cálculos, "d3.stack".

En pantalla muestro un conjunto de datos de juguete que muestra la distribución de uso de un computador de cierto usuario. Despliega la cantidad de minutos dedicados a distintos tipos de actividades, y segmentado por días de la semana.

Bajo ella definimos una función apiladora que retorna "d3.stack". **Esta función espera recibir un arreglo de datos, y a partir de ellos generará una serie de sumas de valores de forma acumulada.**

Para que lo logre, es necesario indicarle cómo realizar los distintos pasos de valores a sumar. Al llamar al método "keys", se le puede indicar un arreglo de llaves que le indican al apilador qué atributos de nuestro *dataset* usar para generar las sumas. En nuestro caso se indican las categorías de uso de tiempo.

Si llamamos a la función resultante con nuestro conjunto de datos y lo imprimimos, podemos ver los cálculos generados. Retorna un arreglo que contiene otros tres arreglos. Si vemos los detalles, vemos que cada uno corresponde a una de las llaves proporcionadas, y que cada una contienen pares de valores numéricos.

El primero contiene pares que comienzan en cero. El segundo, los valores iniciales de cada par son los valores finales del primer arreglo. Lo mismo ocurre entre el tercer arreglo y segundo. Lo que hizo fue sumar en orden de las llaves entregadas los valores correspondientes en el *dataset*, pero recordando los valores finales de la llave anterior.

Esto es perfecto para generar áreas a partir de líneas, **ya que estos arreglos nos entregan los valores verticales de las líneas que forman un área.** Agregaré en el código entonces escalas a partir de los datos generados, para coordenadas en X, en Y y una escala de colores.

Finalmente, definimos una función generadora de área. Lo que hará el programa es generar un *data join* con el resultado del apilador, y a cada arreglo de pares se le llamará esta función, de forma que tendrá acceso a cada par para definir la línea 0 y la línea 1.

Si lo probamos, ¡vemos que funciona! Genera un gráfico de áreas acumuladas curvo en base a un conjunto de datos. Revisa el código en detalle para repasar qué pasó. Como hay tantos arreglos de arreglos es fácil perderse. Recuerda ver que es lo que entra y sale de cada caso, y será más claro.

Bajo la misma idea podemos generar un gráfico de barras apiladas, usando nuestro viejo amigo: el elemento rectángulo. Escribí un ejemplo muy similar al anterior que cambia un par de escalas, pero genera barras apiladas en vez de áreas continuas.

La parte más difícil puede ser el generar los rectángulos, **ya que es necesario dos *data join* sucesivos**. ¡**Revisa el código** y deja tus dudas en el video si tienes alguna!

Con eso termina el contenido de esta cápsula. Recuerda que si tienes preguntas, puedes dejarlas en los comentarios del video para responderlas en la sesión en vivo de esta temática. ¡Chao!