

## Cápsula 4: Marcas en D3.js III

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

Esta la dedicaremos a ver alternativas de generación de **marcas con orientación radial**. Las que revisamos en cápsulas anteriores se pueden utilizar tanto en orientaciones rectilíneas como paralelas, pero ahora hace falta el caso radial.

Lo bueno es que las cosas no cambian tanto en este caso, y por supuesto que D3 provee funciones alternativas para el caso radial. En vez de pensar en eje X y eje Y, debemos pensar en términos de eje angular y eje radial. **Lo que antes trabajábamos como distancia horizontal, lo vemos como distancia angular, y lo que antes era distancia vertical, ahora distancia desde el centro de un círculo, el radio.**

Al seguir esa lógica, D3 provee funciones para definición de líneas y áreas radiales: "lineRadial" y "areaRadial". A modo de ejemplo adapté otro de los ejemplos revisados, con datos con dos atributos.

Una de las diferencias importantes es la definición de escalas, que deben corresponder a este caso. **Ahora una escala se dedica al ángulo dentro de un círculo, y otra al radio, la distancia al centro de un círculo.**

Para el caso de radio, es bastante directo, pero para los ángulos, **D3 trabaja con radianes, no con grados**. Es por eso que al definir una escala de ángulos, definimos el rango como **0 (el inicio de una vuelta al círculo), y dos Pi (el término de una vuelta al círculo)**.

Luego en las definiciones de línea y área, **en vez de métodos "x" e "y", se usan métodos "angle" y "radius"**, que tienen la misma intención que sus correspondencias. Además de eso, no cambia mucho. Las líneas y áreas se dibujan en el origen del sistema coordenado, así que es necesario trasladarlos al generarse con respecto a su centro.

Agregué también un punto y una línea para marcar el centro y el radio máximo de la distribución radial que se está usando en cada caso. Si probamos ejecutando el código, vemos el resultado. **Los ángulos comienzan de la vertical superior y aumentan en sentido horario, mientras que el radio comienza desde el centro y aumenta hacia afuera.**

Finalmente, revisaremos cómo generar arcos circulares. Podemos pensar en inspiración en las secciones de un gráfico de torta que son regiones contiguas que completan un círculo. Al igual que todas las funciones que hemos revisado, D3 provee un generador de arcos llamado "d3.arc".

Este generador se espera sea llamado sobre objetos con propiedades internas que definen el arco. Por ejemplo, comenzamos con un arco que tendrá radio interior cero, radio exterior igual a 100, ángulo inicial 0 y ángulo final igual a un tercio de vuelta.

Si ejecutamos este código, se imprimirá el camino correspondiente en consola. También podemos ver el arco visualmente que agregué en un elemento "path" nuevo.

"d3.arc" también permite personalización. Al generador se le puede especificar un poco de espaciado angular que permite separar arcos sucesivos, mediante "padAngle". También, se pueden redondear las esquinas de los arcos generados en cierta cantidad. Si ejecutamos estos cambios, vemos que se suaviza el arco generado.

Si probamos generando otros arcos con otras propiedades geométricas, podemos ver el círculo completo que efectivamente tiene espacio entre arcos.

También es posible establecer alguna de estas propiedades geométricas como constantes entre todos los arcos generados. En nuestro caso, podríamos definir un generador los valores de radio interno y externo, que tendrán los arcos que agregaremos de forma separada después. Si lo probamos, vemos el resultado de establecer un radio interno distinto de cero.

Esta interfaz para generar arcos es conveniente para realizar gráficos de torta, el problema es que debemos generar los objetos de geometría de estos arcos primero, y esto requiere computación angular a partir de datos.

Pero como podrán suponer, D3 provee una función que ayuda con esto, "d3.pie". **Esta es similar a "d3.stack" en que se usa para preprocesar datos y llevarlo a algo que D3 puede ocupar para dibujar.**

En este caso, espera recibir un arreglo de datos y una forma de determinar el valor a codificar mediante ángulo. A partir del valor especificado en el método "value", determina el valor total correspondiente al círculo completo y calculará los arcos correspondientes a cada dato en base a eso.

Al llamarse con un conjunto de datos de juguete, produce objetos que una función de arcos puede procesar y dibujar después. Si definimos una función de arcos y agregamos un elemento "path" para cada arco generado, podemos crear los arcos visualmente.

Algo conveniente de los arcos generados por "d3.pie" es que transfiere el dato correspondiente al arco en un atributo "data", así podemos acceder al dato original desde el arco y hacer personalizaciones a partir de eso.

Si probamos esto, vemos un gráfico de torta fiel a los datos entregados. Habrás notado que definimos un método "sort" de "d3.pie" como nulo. Esto generó que el orden de los arcos sea idéntico al orden especificado en los datos. Si quitamos esa definición, por defecto se ordenarán los arcos desde el más largo al más corto.

Con eso termina el contenido de esta cápsula y las de este grupo. Vimos muchas, y casi todas, las funciones de generaciones geométricas de D3. Aún así, todos estos métodos presentan más detalles que permiten personalizar aún más las formas o cubrir casos que no revisamos. **Te invito a revisarlas en “d3-shape” y jugar con los ejemplos que vimos.**

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