

Cápsula 1: d3-brush y panorámica-detalle

Hola, bienvenidxs a una cápsula del curso Visualización de Información. En esta revisaremos ejemplos de visualizaciones interactivas usando D3.js, específicamente d3-brush, y lo aplicaremos para generar una visualización con *facet* de panorámica-detalle.

Comenzaremos con [“d3-brush”](#). Este es un sub paquete de D3.js que provee objetos con comportamiento interactivo listo. Sigue una forma similar de declarar objetos a la de “d3-zoom”, que revisamos, pero aquí se generan “brushes”, que permiten crear selecciones de regiones en una o dos dimensiones.

Esto lo hace creando una capa de interacción sobre una región especificada, que está atenta a eventos gatillados por interacción de usuario. Agregaremos una capa de selección sobre nuestro ejemplo básico de gráfico de dispersión, en la parte que contiene puntos.

Para eso creamos una forma básica de este comportamiento, con el método “brush”, como nuestro en pantalla. Con el método “extent” se especifica las coordenadas relativas de la región a la que se puede interactuar. Como quiero que solo se aplique sobre la región con puntos de nuestro gráfico, le entrego las dimensiones de esa región.

Luego, hay que llamar este comportamiento sobre algún contexto, de forma similar a como lo hicimos con “d3-zoom”. En este caso, creé un contenedor separado para nuestro “brush”, que está trasladado de igual forma que la región de los puntos. Con un llamado del comportamiento, mediante “call” sobre este grupo, D3 hace toda la magia de agregar elementos al DOM que permiten la interacción.

Si lo probamos, al pasar el cursor sobre la región inmediatamente podemos ver que está algo distinto tiene. Al cliquear y arrastrar el cursor, vemos que una región cuadrada nueva aparece. No solo eso, sino que se puede arrastrar la región creada. Si te fijas, solo permite moverla entre los límites definidos y especificados en el método “extent”. También permite reescalar esta selección o región arrastrando sus bordes.

Si inspeccionamos el SVG, vemos que dentro del contenedor creado se agregan un montón de elementos invisibles con eventos. Cada uno está destinado a alguna acción distinta, de las que vimos recién.

Además de todo eso, podemos asociar código que se gatille cada vez que una región de selección se genere. Podemos entregar una función mediante “on” en el objeto de interacción, junto al evento “brush”. Esta función recibirá un objeto de evento con información de la acción, como las coordenadas de la selección realizada. Si lo probamos, podemos ver impresiones de estas coordenadas.

Esto es bastante poderoso, ya que ahora al contar con las coordenadas, podemos realizar efectos sobre nuestra visualización. Lo que haré será cambiar el color de todos los puntos que se encuentren dentro del rango de la región actual.

El código en pantalla realiza esto. Como la selección tiene coordenadas geométricas del DOM, e internamente los puntos están asociados a los datos con valores originales, necesito obtener rangos de valores que corresponden a esa selección. Para eso, utilizo las escalas usadas para definir los ejes, y su función "invert" que transforma un dato del rango a un dato de su dominio.

Con eso obtengo un rango de valores en los atributos originales "x" e "y" de los datos, y puedo crear una función de filtrado, que determina si es que un dato efectivamente está en ese rango o no. Así, finalmente se actúa sobre la selección de puntos del gráfico, y dependiendo de si el dato asociado a cada punto está en el rango, se colorea magenta con opacidad 1, o gris con baja opacidad.

Si lo probamos, veremos que funciona, se capturan correctamente los puntos del rango asociado y se colorean de forma esperada. Ahora, esta capa de interacción provee una forma muy específica de generar selección de una región, con muchas opciones de cambios que tal vez no deseamos.

Lo bueno es que todo esto es personalizable. Una forma es modificando los elementos SVG generados por el comportamiento, y otras es mediante métodos que provee "d3-brush". En pantalla veremos varias modificaciones, usando los métodos "filter" sobre el comportamiento, llamando el método "move" y modificando contenido dentro del contenedor con la capa generada.

Si lo probamos, vemos varios cambios. La selección comienza en pantalla, tiene fondo magenta, y varias opciones de interacción ya no funcionan, y solo permiten mover esta región acotada de selección. Investiga la documentación de "d3-brush" e identifica qué hace cada sentencia usada para lograr esto: "filter", "move" y los cambios dentro del DOM. ¡Deja en los comentarios lo que averiguaste!

Pensando en el reciente contenido de *facet*, este tipo de interacción recuerda al *idiom* de *bird-eye*, que permite ver el detalle de una región seleccionada dentro de un área extensa. De hecho, con lo que ya revisamos en nuestro gráfico de dispersión, no es tan difícil extenderlo para generar un *facet* de partición de datos como panorámica-detalle.

En pantalla muestro el resultado de un intento de esto, donde la vista de la derecha se ve el detalle de la selección a la izquierda. El código necesario para esto no es tan complejo, era necesario generar otro espacio para una segunda vista, y vincular en el llamado de "brush" para que se actualice dicha vista. Puedes revisar el código de este ejemplo para ver cómo funciona.

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!