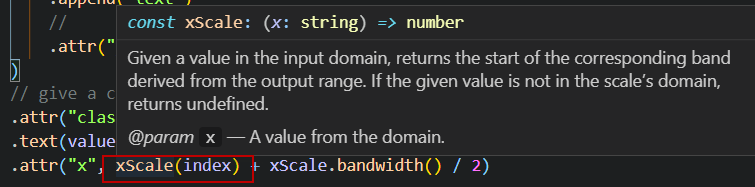
05 interactivity tool tip on bar chart

<https://www.youtube.com/watch?v=BS5VU0gFgE0&t=6s>

Display the value of the bar chart on hover over it

Positioning the scales

xScale – takes the index of the current bar element



yScale – uses the value of the current bar

Commented code from video

///https://www.youtube.com/watch?v=BS5VU0gFgE0&t=6s

import React, { useRef, useEffect, useState } from "react";

import "./App.css";

import { select, axisBottom, scaleLinear, axisRight, scaleBand } from "d3";

*function* App() {

*const* [data, setData] = useState([25, 30, 45, 60, 10, 65, 75]);

*const* svgRef = useRef();

  useEffect(() *=>* {

*const* svg = select(svgRef.current);

*const* xScale = scaleBand()

      .domain(data.map((*value*, *index*) *=>* index))

      .range([0, 300])

      .padding(0.5);

*const* yScale = scaleLinear().domain([0, 150]).range([150, 0]);

*const* colorScale = scaleLinear()

      .domain([75, 100, 150])

      .range(["green", "orange", "red"])

      .clamp(true);

*const* xAxis = axisBottom(xScale).ticks(data.length);

    svg.select(".x-axis").style("transform", "translateY(150px)").call(xAxis);

*const* yAxis = axisRight(yScale);

    svg.select(".y-axis").style("transform", "translateX(300px)").call(yAxis);

    svg

      .selectAll(".bar")

      .data(data)

      .join("rect")

      .attr("class", "bar")

      .style("transform", "scale(1, -1)")

      .attr("x", (*value*, *index*) *=>* xScale(index))

      .attr("y", -150)

      .attr("width", xScale.bandwidth())

      // add onto the chain before the transition

      // need to define the event and what to do when the event triggers

      // the on mouse enter's handler function recives the value and index of the data array

      .on("mouseenter", (*value*, *index*) *=>* {

        // every time on mouse enter do this stuff

        svg

          .selectAll(".tooltip")

          // sync with value from the data array of just that bar

          .data([value])

          // for every entering and updating piece  of data

          .join((*enter*) *=>*

            enter

              //show the text of the value

              .append("text")

              // give them the initial y scale attribute

              // this makes the annimation of the y axis on line 75

              // just go from 4 to 8 instead of 0 to 8

              .attr("y", yScale(value) - 4)

          )

          // give a class of tooltip

          .attr("class", "tooltip")

          .text(value)

          /// position the text element on the x axis

          // xScale takes in the index of the selected bar element

          // this is the index from the item in the data array,

          // not the index from .data([value]) on line 48 which would be 0

          // adding 1/2 the bandwidth  pushes the text over to help center it

          .attr("x", xScale(index) + xScale.bandwidth() / 2)

          // this will center the text element on the left edge of bar

          .attr("text-anchor", "middle")

          // adding a transition to the the y positioning

          .transition()

          // postion the element on the y axis

          // yScale will use the value of the current bar

          // puts the text element on each bar -8 gives it a bit of margin

          .attr("y", yScale(value) - 8)

          //trainsitioning the opacity of the text element when it comes in

          .attr("opacity", 1);

      })

      // on mouse leave select the elements with the class .tooltip and remove that class

      .on("mouseleave", () *=>* svg.select(".tooltip").remove())

      .transition()

      .attr("fill", colorScale)

      .attr("height", (*value*) *=>* 150 - yScale(value));

  }, [data]);

  return (

    <React.Fragment>

      <svg ref={svgRef}>

        <g className="x-axis"></g>

        <g className="y-axis"></g>

      </svg>

      <br />

      <br />

      <br />

      <button onClick={() *=>* setData(data.map((*value*) *=>* value + 5))}>

        Update data

      </button>

      <button onClick={() *=>* setData(data.filter((*value*) *=>* value < 35))}>

        Filter data

      </button>

      <button

        onClick={() *=>* setData([...data, Math.round(Math.random() \* 100)])}

      >

        Add Data

      </button>

    </React.Fragment>

  );

}

export default App;

