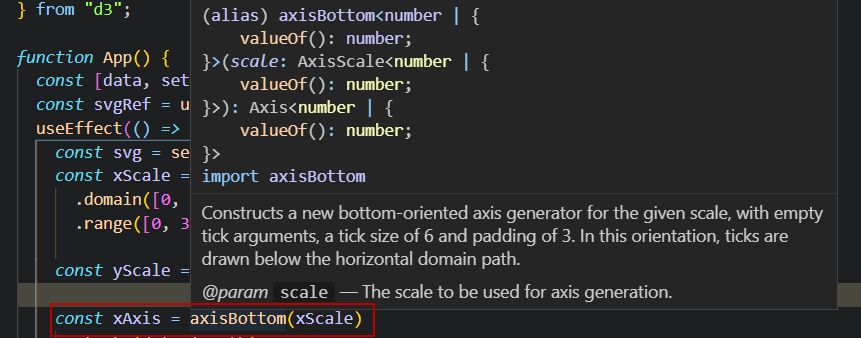
03 React hooks line chart axes and scales

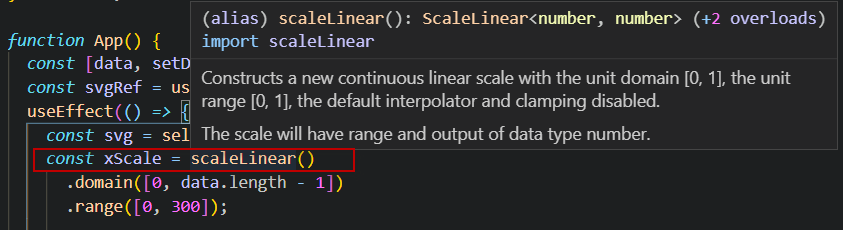
The line function helps to shape the path element

Will need to import axisBottom, scaleLinear and axisRight

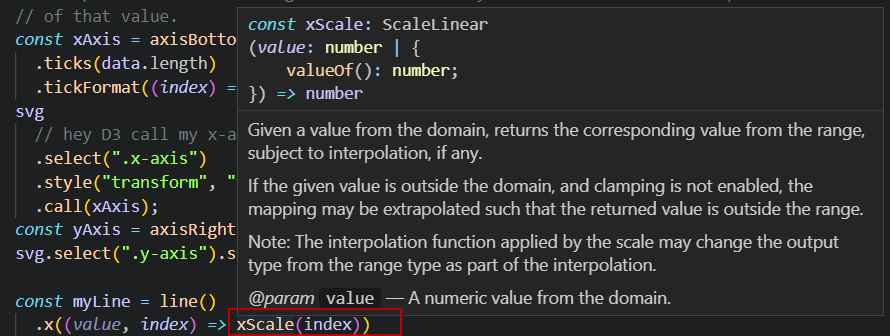
axisBottom expects a scale



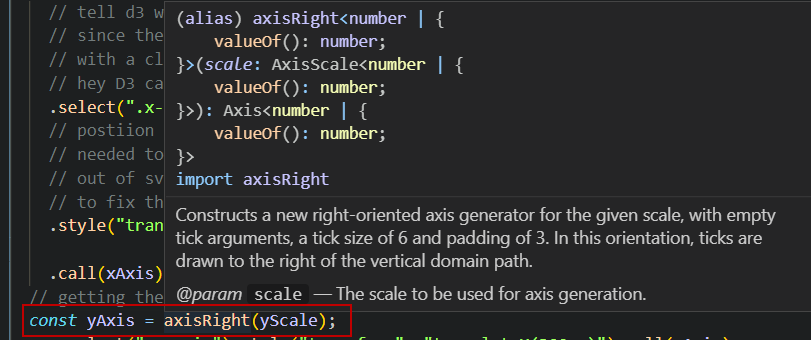
Using scaleLinear() to make the xScale const – it needs a domain and range



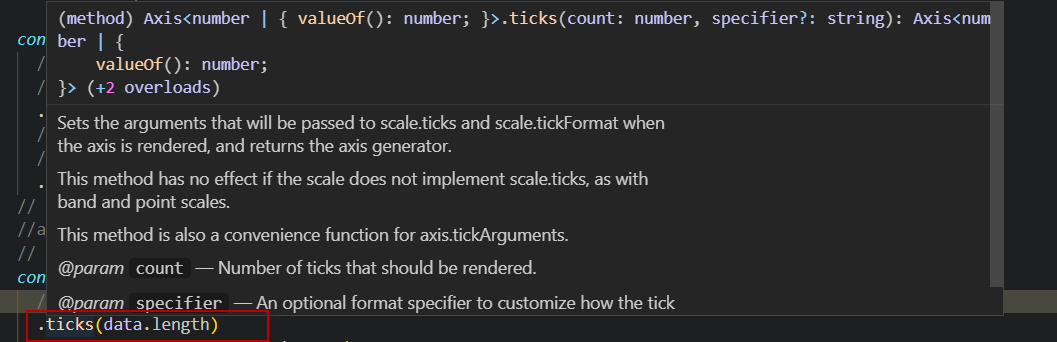
xScale takes in the index



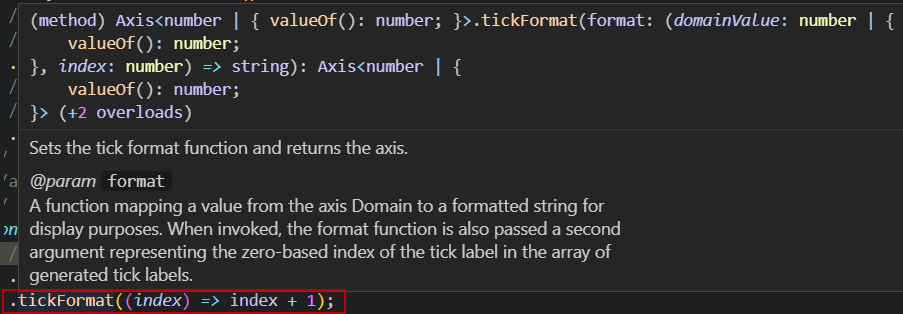
axisRight



D3 Ticks method



D3 Tick formater



Commented code

// https://www.youtube.com/watch?v=yhwHUmjqxQw

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

import "./App.css";

import {

  select,

  line,

  curveCardinal,

  axisBottom,

  scaleLinear,

  axisRight,

} from "d3";

*function* App() {

*const* [data, setData] = useState([25, 30, 45, 60, 20, 65, 75]);

*const* svgRef = useRef();

  useEffect(() *=>* {

*const* svg = select(svgRef.current);

*const* xScale = scaleLinear()

      // takes in input values

      // the domain here is the index values of the data array lenght-1

      // scale number of elements in the array up to the range

      .domain([0, data.length - 1])

      // range is the visual representaiton of the data in the array

      .range([0, 300]);

    // make the yScale

*const* yScale = scaleLinear()

      // the domain for the y scale is input values

      // of 0 to 150

      .domain([0, 150])

      // the range here is output values

      // 150 to 0

      .range([150, 0]);

    // axisBottom expects a scale- which is a function or helper that transforms

    //an input value to something else that is usualy needed for the visual representation

    // of that value.

*const* xAxis = axisBottom(xScale)

      // add the correct amount of ticks to the axis

      // using the length of data array

      .ticks(data.length)

      // values from index +1 so that we see 1-7 not 0 to 6

      .tickFormat((*index*) *=>* index + 1);

    svg

      // tell d3 where to position the axis in the svg

      // since there is only one x and y create a group  g element in the SVG with

      // with a class of x-axis and y-axis

      // hey D3 call my x-axis into the g element with this class name

      .select(".x-axis")

      // postiion the x-axis to the bottom of the SVG

      // needed to add overflow visible to see at frist becuase the line moved

      // out of svg - the lines,path and ticks are being created in the g element by D3

      // to fix this needed to change the selectAll to .line not path in the svg selection belodw

      .style("transform", "translateY(150px)")

      .call(xAxis);

    // getting the axis on the Right

    // pass in the yScale

*const* yAxis = axisRight(yScale);

    // create a new group in the SVG with a class of y-axis

    svg

      .select(".y-axis")

      // move to the very right of the SVG

      .style("transform", "translateX(300px)")

      //pass in the yAxis

      .call(yAxis);

*const* myLine = line()

      // pass index value into scale

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

      // pass in the y scale defined as scaleLinear() previoulsy

      .y(yScale)

      .curve(curveCardinal);

    svg

      // selectAll to .line not path

      .selectAll(".line")

      .data([data])

      .join("path")

      // attahch the class of line to new and updating elements

      .attr("class", "line")

      // passing in myLine here

      .attr("d", myLine)

      .attr("fill", "none")

      .attr("stroke", "blue");

  }, [data]);

  return (

    <React.Fragment>

      <svg ref={svgRef}>

        <g className="x-axis"></g>

        <g className="y-axis"></g>

      </svg>

      <br />

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

        Update data

      </button>

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

        Filter data

      </button>

    </React.Fragment>

  );

}

export default App;