CS 405 ASSIGNMENT 1 - ŞİMAL YÜCEL 29420

I chose my data to portray and created 3 files: html, js and css. I first started by writing my html file. I specified a SVG bar chart and linked it to my js file. I also linked it to my css file for stylesheet.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SVG Bar Chart</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <svg id="bar-chart" width="800" height="600" ></svg>
    <script src="java.js"></script>
</body>
</html>
```

Then I defined my css file for the <body> element of my bar chart, to choose the background color and centralize everything etc.

```css
body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f4f4f4;
}
```

Then I started writing my js file. Because we couldnt use libraries I manually entered the data from 2015-2022. after that I did the configurations for my SVG and the bars. And I calculated the max suicide number and crude suicide rate to conclude a range for my right and left axises.

```javascript
// Configuration for the chart
const svgWidth = 800;
const svgHeight = 600;
const barWidth = 60; // width of each bar
const barSpacing = 20; // space between bars
const maxBarHeight = 500; // maximum height a bar can take up in the SVG
const yAxisPadding = 50; // space to leave at the bottom for year labels



const maxSuicides = Math.max(...data.map(d => d.no_of_suicides));
const scalingFactor = maxBarHeight / maxSuicides;

const maxRate = Math.max(...data.map(d => d.crude_suicide_rate));
const scalingFactor2 = maxBarHeight / maxRate;
```

After that I defined my right and left axises, with the element ‘line’. I defined the coordinates (xs and ys) for where they should stand and I added and subtracted numbers manually to see where they stand and finalized it on my liking. I also defined their color and width.

```
// Reference to SVG
const svg = document.getElementById('bar-chart');

const yAxis = document.createElementNS('http://www.w3.org/2000/svg', 'line');
yAxis.setAttribute('x1', yAxisPadding -10); // A bit to the left from where my bars start
yAxis.setAttribute('y1', svgHeight - yAxisPadding+40);
yAxis.setAttribute('x2', yAxisPadding-10 );
yAxis.setAttribute('y2', 60);
yAxis.setAttribute('stroke', 'black');
yAxis.setAttribute('stroke-width', '2');
svg.appendChild(yAxis);

const rightYAxisPadding = 50;
const rightYAxisXPos = svgWidth - rightYAxisPadding + 10;

const rightYAxis = document.createElementNS('http://www.w3.org/2000/svg', 'line');
rightYAxis.setAttribute('x1', rightYAxisXPos-20);
rightYAxis.setAttribute('y1', svgHeight - yAxisPadding + 40);
rightYAxis.setAttribute('x2', rightYAxisXPos-20);
rightYAxis.setAttribute('y2', 60);
rightYAxis.setAttribute('stroke', 'black');
rightYAxis.setAttribute('stroke-width', '2');
svg.appendChild(rightYAxis);
```

Then for my lines, I declared an interval for a for loop to iterate to max suicide rate and create ticks and assign number labels to the ticks. For each iteration of the loop, a yPosition for the tick mark and label is calculated. The value is determined based on the height of the SVG (svgHeight), the current value of i (multiplied by a scalingFactor), and a padding value (yAxisPadding). I also did the same approach for the right axis with a different interval because the numbers were smaller.

```
const interval = 1000;
for (let i = 0; i <= maxSuicides; i += interval) {
    const yPosition = svgHeight - (i * scalingFactor) - yAxisPadding;

    // Small tick mark
    const tick = document.createElementNS('http://www.w3.org/2000/svg', 'line');
    tick.setAttribute('x1', yAxisPadding - 15);
    tick.setAttribute('y1', yPosition);
    tick.setAttribute('x2', yAxisPadding +10);
    tick.setAttribute('y2', yPosition);
    tick.setAttribute('stroke', 'black');
    tick.setAttribute('stroke-width', '2');
    svg.appendChild(tick);

    // Number label
    const numberLabel = document.createElementNS('http://www.w3.org/2000/svg', 'text');
    numberLabel.setAttribute('x', yAxisPadding - 50);
    numberLabel.setAttribute('y', yPosition + 5);
    numberLabel.textContent = i;
    svg.appendChild(numberLabel);
}
```

I also assigned texts to these lines to show what they stand for which you can see in the continuing code where I used ‘test’ element, and played with x and y axises to find a right position for them.

I actually did this part first but it is after all these codes so I am mentioning it now. This is the code for the bars. The for loop goes through my data. The height of the bar (barHeight) is determined by multiplying the number of suicides in the given year (datum.no_of_suicides) by a scaling factor (scalingFactor). This ensures the bar visually represents the data correctly within the SVG. A label is created below each bar to display the year (datum.year) the data corresponds to. Another label is created to display inside the bar, which shows the exact number of suicides (datum.no_of_suicides).

```
data.forEach((datum, index) => {
    // Calculate x position, height and scaled height for the bar
    const x = yAxisPadding + index * (barWidth + barSpacing) + barSpacing;

    const barHeight = datum.no_of_suicides * scalingFactor;
    // Calculate y position for the bar (it starts from the top in SVG, so need to invert)
    const y = svgHeight - barHeight - yAxisPadding;
    // Create the bar element
    const bar = document.createElementNS('http://www.w3.org/2000/svg', 'rect');
    bar.setAttribute('x', x);
    bar.setAttribute('y', y);
    bar.setAttribute('width', barWidth);
    bar.setAttribute('height', barHeight);
    bar.setAttribute('fill', '#3498db'); // you can set any color you prefer for the bars

    // Append the bar to the SVG
    svg.appendChild(bar);

    // Create label for the year below the bar
    const yearLabel = document.createElementNS('http://www.w3.org/2000/svg', 'text');
    yearLabel.setAttribute('x', x + barWidth / 2);
    yearLabel.setAttribute('y', svgHeight - 20);
    yearLabel.setAttribute('text-anchor', 'middle'); // to center the text
    yearLabel.textContent = datum.year;
    svg.appendChild(yearLabel);

    // Add data value inside the bar
const valueLabel = document.createElementNS('http://www.w3.org/2000/svg', 'text');
valueLabel.setAttribute('x', x + barWidth / 2);
valueLabel.setAttribute('y', y + 20);
valueLabel.setAttribute('text-anchor', 'middle');
valueLabel.setAttribute('fill', 'white'); // set font color to white for visibility against bar color
valueLabel.textContent = datum.no_of_suicides;
svg.appendChild(valueLabel);
});
```

Then I needed to create a red line like on the site for a better visualization and to show the crude suicide rate. A string, polylinePoints, is populated with x and y coordinates based on each data entry's crude_suicide_rate. An SVG polyline is created using these coordinates. For each data entry, A red circle (data point) is plotted above the center of its respective bar, based on crude_suicide_rate. A label, displaying the suicide rate value, is positioned above its respective circle.

```
let polylinePoints = "";
data.forEach((datum, index) => {
    const x = yAxisPadding+ index * (barWidth + barSpacing) + barSpacing + (barWidth / 2);  // center of the bar
    const rateHeight = datum.crude_suicide_rate * rateScalingFactor;
    const y = svgHeight - rateHeight - yAxisPadding;

    polylinePoints += `${x},${y} `;
});

const polyline = document.createElementNS('http://www.w3.org/2000/svg', 'polyline');
polyline.setAttribute('points', polylinePoints.trim());
polyline.setAttribute('fill', 'none');
polyline.setAttribute('stroke', 'red');
polyline.setAttribute('stroke-width', '2');
svg.appendChild(polyline);

data.forEach((datum, index) => {
    const x = yAxisPadding+ index * (barWidth + barSpacing) + barSpacing + (barWidth / 2);
    const rateHeight = datum.crude_suicide_rate * rateScalingFactor;
    const y = svgHeight - rateHeight - yAxisPadding;

    // Create the circle for each data point
    const circle = document.createElementNS('http://www.w3.org/2000/svg', 'circle');
    circle.setAttribute('cx', x);
    circle.setAttribute('cy', y);
    circle.setAttribute('r', '5'); // radius of the circle
    circle.setAttribute('fill', 'red');
    svg.appendChild(circle);

    // Add text label above the circle to show the value
    const rateLabel = document.createElementNS('http://www.w3.org/2000/svg', 'text');
    rateLabel.setAttribute('x', x);
    rateLabel.setAttribute('y', y - 10);
    rateLabel.setAttribute('text-anchor', 'middle'); // to center the text
    rateLabel.setAttribute('font-size', '12px');
    rateLabel.textContent = datum.crude_suicide_rate.toFixed(2);
    svg.appendChild(rateLabel);
});
```

Lastly to put it all together at the end of my code, I created a blue square and a red line legend to show what everything indicates in the graph; using the similar approaches I used though all of my code. I played with the numbers a lot to decide where I should locate everything. At last, my end result looked like this: