CS 405 Assignment 1

Selim Gül

29200

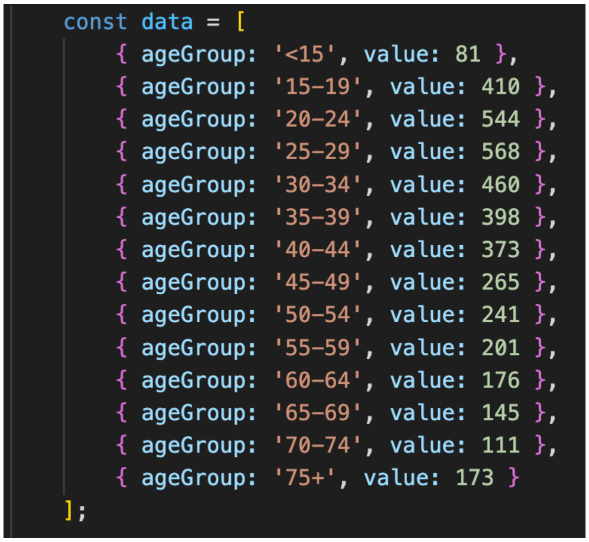
Fall 2023

Introduction

In this assignment, I have visualized a data which is available on the TUIK website using SVG. The data I chose to visualize was *İstatistiki Bölge Birimleri Sınıflaması ve yaş grubuna göre intiharlar, 2002-2022* (Suicides by Statistical Regions and age group, 2002-2022).

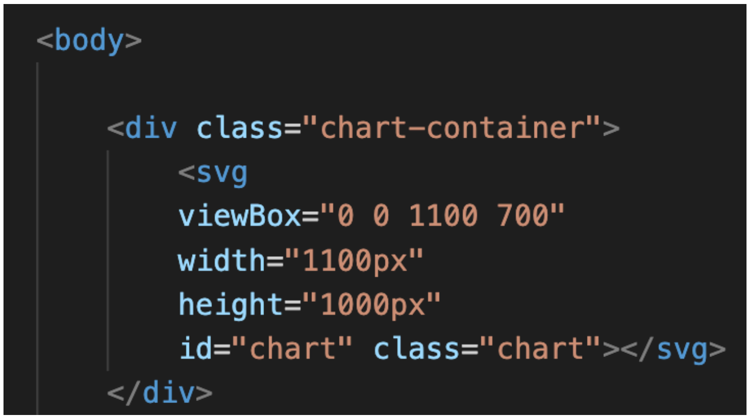
Visualization and Methodology

To visualize this data, first, I created an html file called *index.html.* I did not create any seperate CSS or JS files, therefore I wrote all my code as inline. The initial phase involved getting the data from TUIK and hard-coding it.



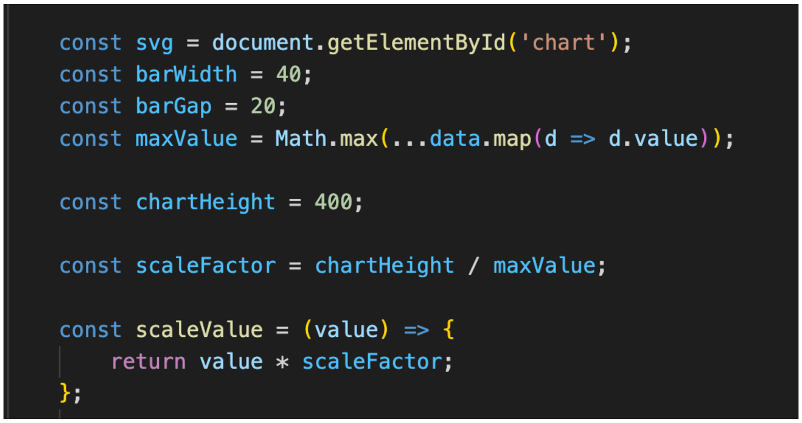
```
const data = [
    { ageGroup: '<15', value: 81 },
    { ageGroup: '15-19', value: 410 },
    { ageGroup: '20-24', value: 544 },
    { ageGroup: '25-29', value: 568 },
    { ageGroup: '30-34', value: 460 },
    { ageGroup: '35-39', value: 398 },
    { ageGroup: '40-44', value: 373 },
    { ageGroup: '45-49', value: 265 },
    { ageGroup: '50-54', value: 241 },
    { ageGroup: '55-59', value: 201 },
    { ageGroup: '60-64', value: 176 },
    { ageGroup: '65-69', value: 145 },
    { ageGroup: '70-74', value: 111 },
    { ageGroup: '75+', value: 173 }
];
```

Then, I created a svg element inside a div called “chart-container”, and set the width of the SVG to 1100px, and height to 1000px.



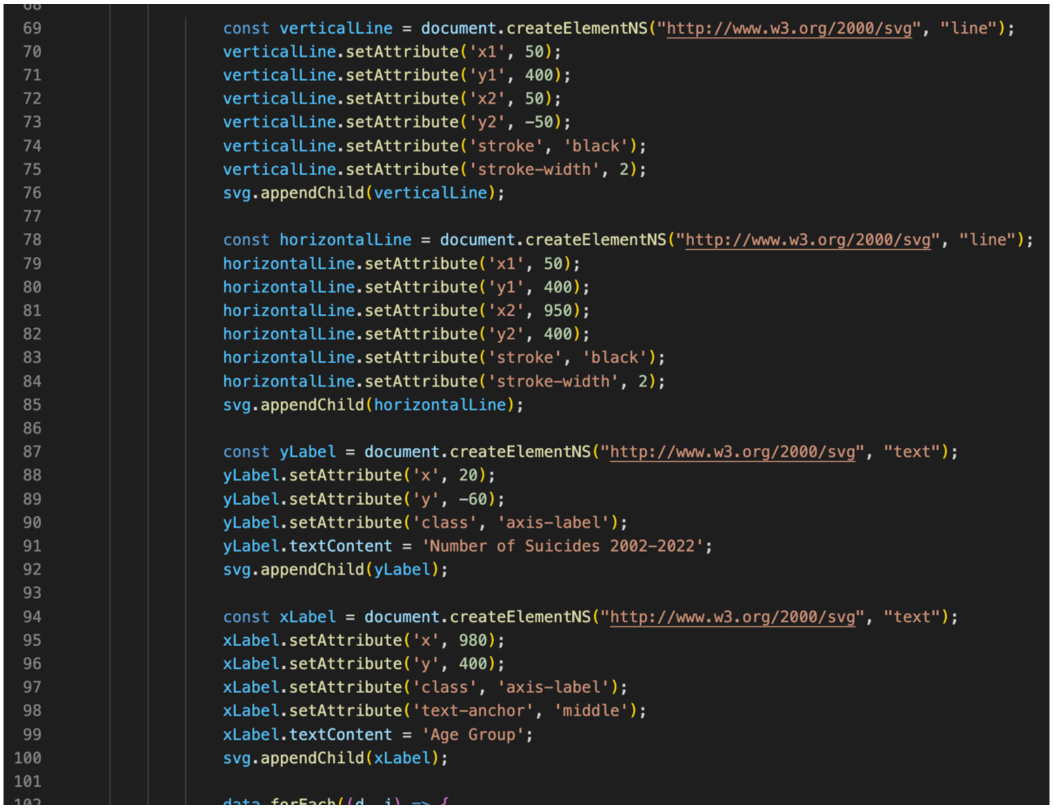
```
<body>

    <div class="chart-container">
        <svg
        viewBox="0 0 1100 700"
        width="1100px"
        height="1000px"
        id="chart" class="chart"></svg>
    </div>
```

Afterwards, I initialized the svg in JS, along with a maximum height value that a bar can take, widths, and gaps between the two bars. The scaleValue here is important to make sure that the number attached to a bar doesn’t exceed the number on the y axis of the chart. So, it helps to scale the height of the value bars according to the height of the chart overall.



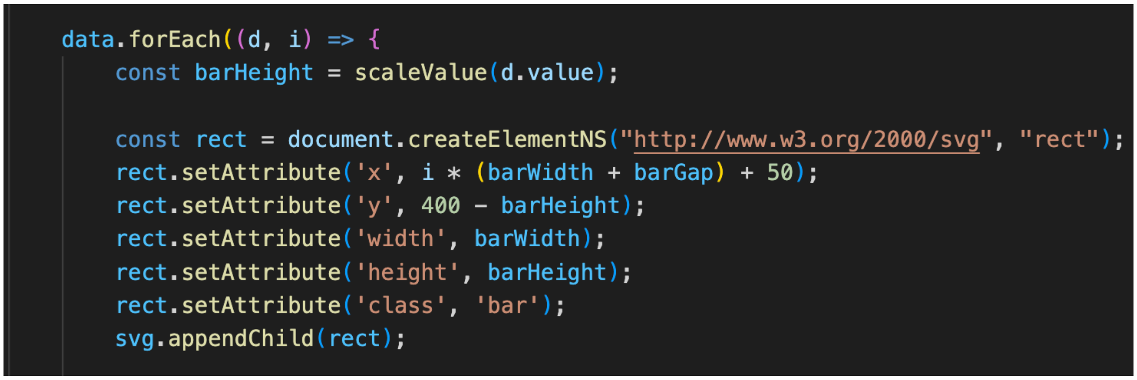
```
const svg = document.getElementById('chart');
const barWidth = 40;
const barGap = 20;
const maxValue = Math.max(...data.map(d => d.value));

const chartHeight = 400;

const scaleFactor = chartHeight / maxValue;

const scaleValue = (value) => {
    return value * scaleFactor;
};
```

Later, I created the axises and the labels for them using “line” and “text” properties. The y Axis label is positioned at (20,-60) to make sure that it doesn’t overlap with the line, and the same for x axis label which is placed at (980, 400)



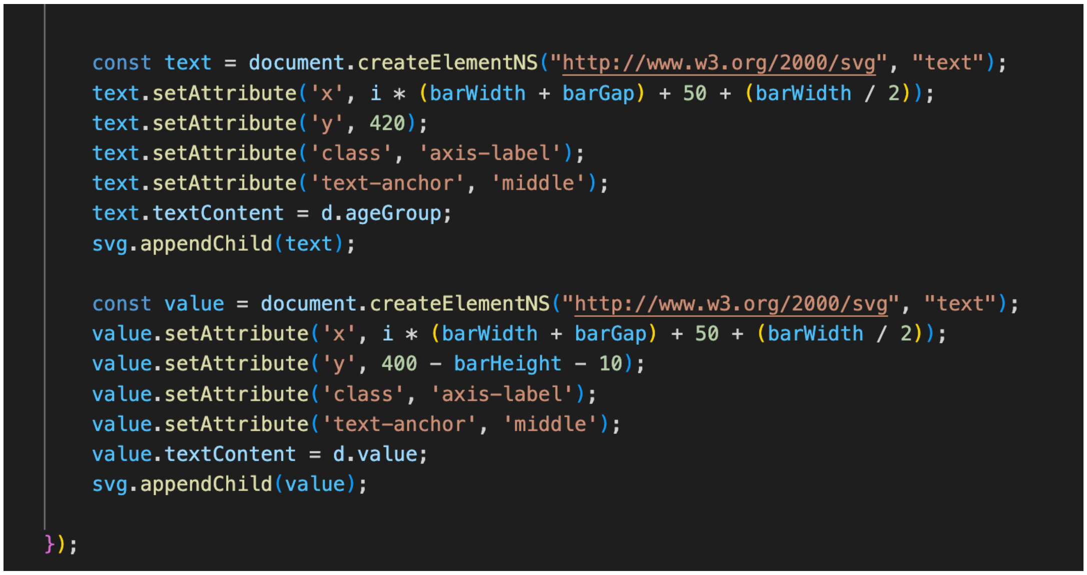
```
            const verticalLine = document.createElementNS("http://www.w3.org/2000/svg", "line");
            verticalLine.setAttribute('x1', 50);
            verticalLine.setAttribute('y1', 400);
            verticalLine.setAttribute('x2', 50);
            verticalLine.setAttribute('y2', -50);
            verticalLine.setAttribute('stroke', 'black');
            verticalLine.setAttribute('stroke-width', 2);
            svg.appendChild(verticalLine);

            const horizontalLine = document.createElementNS("http://www.w3.org/2000/svg", "line");
            horizontalLine.setAttribute('x1', 50);
            horizontalLine.setAttribute('y1', 400);
            horizontalLine.setAttribute('x2', 950);
            horizontalLine.setAttribute('y2', 400);
            horizontalLine.setAttribute('stroke', 'black');
            horizontalLine.setAttribute('stroke-width', 2);
            svg.appendChild(horizontalLine);

            const yLabel = document.createElementNS("http://www.w3.org/2000/svg", "text");
            yLabel.setAttribute('x', 20);
            yLabel.setAttribute('y', -60);
            yLabel.setAttribute('class', 'axis-label');
            yLabel.textContent = 'Number of Suicides 2002-2022';
            svg.appendChild(yLabel);

            const xLabel = document.createElementNS("http://www.w3.org/2000/svg", "text");
            xLabel.setAttribute('x', 980);
            xLabel.setAttribute('y', 400);
            xLabel.setAttribute('class', 'axis-label');
            xLabel.setAttribute('text-anchor', 'middle');
            xLabel.textContent = 'Age Group';
            svg.appendChild(xLabel);

            data.forEach((d, i) => {
```

For drawing the bar charts, I used a for loop instead of hard-coding each bar. My loop iterates over each value appended to the set *data.* Here, the above-mentioned scaleValue is used to determine the height of the bars. Their position is calculated using the following logic: x axis position is determined by the width of the bar (40) + gap between the bars (20) multiplied by the order of the bar, and added 50 as default to not place the first bar at (0,0) on the graph. The y axis location is fairly simple, the height of the chart (400) minus the height of the bar, so the starting point of the bar is placed at the x axis of the chart.



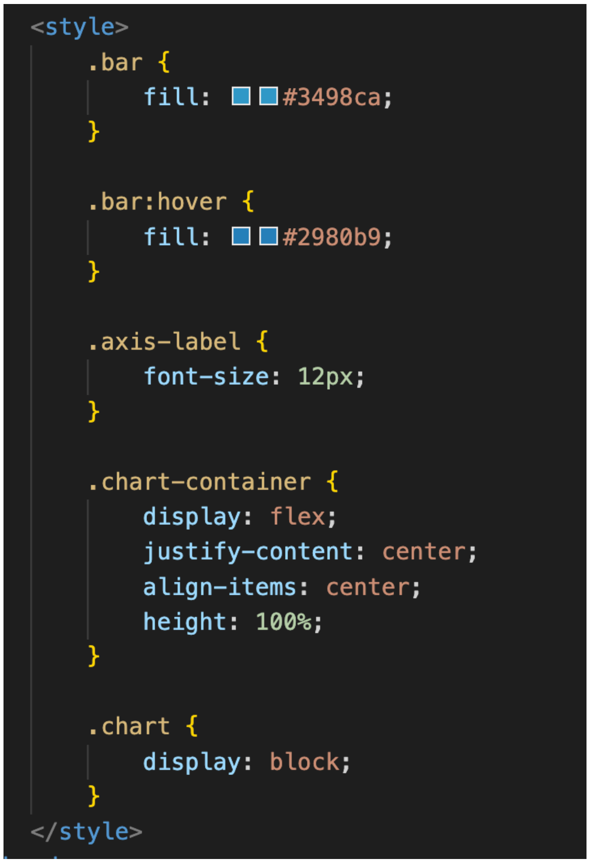
```
data.forEach((d, i) => {
    const barHeight = scaleValue(d.value);

    const rect = document.createElementNS("http://www.w3.org/2000/svg", "rect");
    rect.setAttribute('x', i * (barWidth + barGap) + 50);
    rect.setAttribute('y', 400 - barHeight);
    rect.setAttribute('width', barWidth);
    rect.setAttribute('height', barHeight);
    rect.setAttribute('class', 'bar');
    svg.appendChild(rect);
```

To create the value labels, the same logic above is applied within the same for loop.



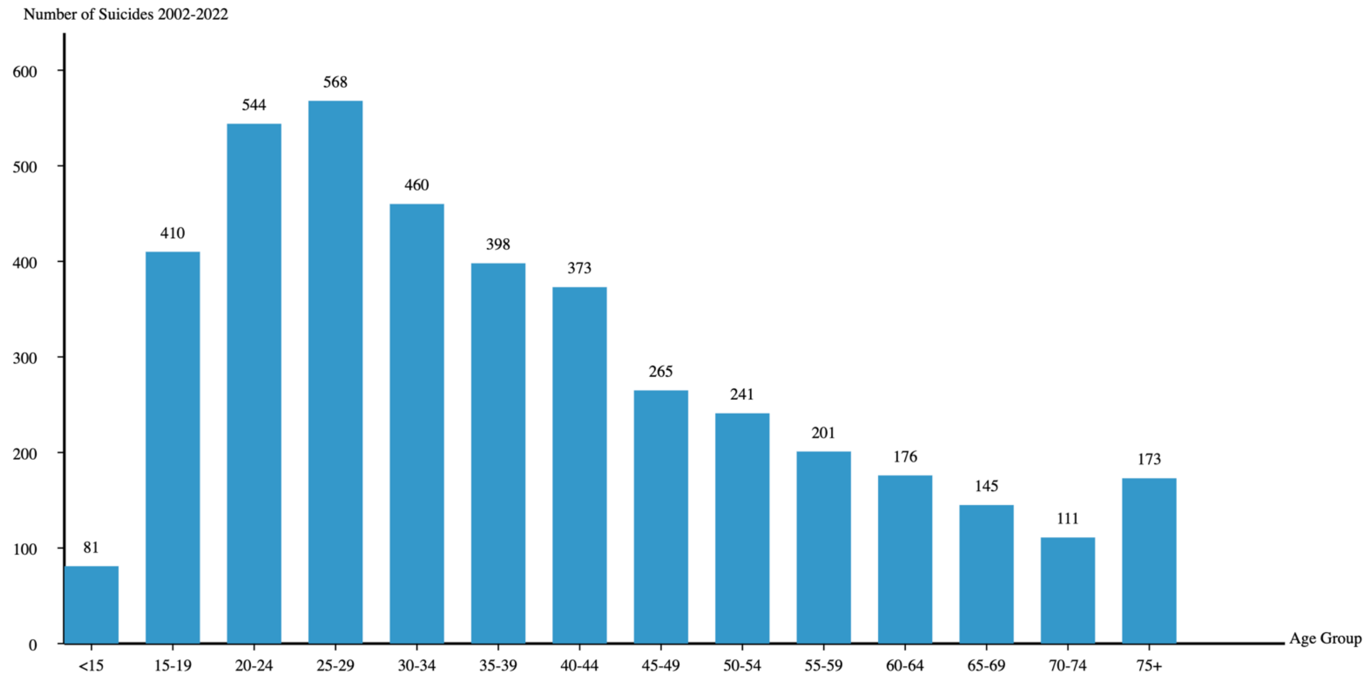
```
    const text = document.createElementNS("http://www.w3.org/2000/svg", "text");
    text.setAttribute('x', i * (barWidth + barGap) + 50 + (barWidth / 2));
    text.setAttribute('y', 420);
    text.setAttribute('class', 'axis-label');
    text.setAttribute('text-anchor', 'middle');
    text.textContent = d.ageGroup;
    svg.appendChild(text);

    const value = document.createElementNS("http://www.w3.org/2000/svg", "text");
    value.setAttribute('x', i * (barWidth + barGap) + 50 + (barWidth / 2));
    value.setAttribute('y', 400 - barHeight - 10);
    value.setAttribute('class', 'axis-label');
    value.setAttribute('text-anchor', 'middle');
    value.textContent = d.value;
    svg.appendChild(value);

});
```

And finally, to add a couple of styling properties like *hover* and to make the chart displayed in the middle of the screen, couple of lines of CSS code was used.



```
<style>
    .bar {
        fill: #3498ca;
    }

    .bar:hover {
        fill: #2980b9;
    }

    .axis-label {
        font-size: 12px;
    }

    .chart-container {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100%;
    }

    .chart {
        display: block;
    }
</style>
```

The outcome of the implementation is the following:


Number of Suicides 2002-2022
<15
Age Group