The website link - Website to view the current version of the visualization.
The current version of the code - Link to the assignment Github.

Introduction

The differences in health outcomes between various socioeconomic groups have drawn more attention in recent years. Income inequality is one of the most important causes causing these differences. Research indicates that nations with higher levels of income disparity frequently have worse health outcomes, including lower life expectancy and higher rates of infant mortality. Using data from 2021, this research intends to investigate the relationship among OECD nations between infant mortality, life expectancy, and income inequality. The research aims to visualize this data in order to identify trends that illustrate the ways in which income inequality affects population health.

Project Objectives

Examine how health outcomes are impacted by income inequality: To inquire the correlation between important health outcomes like life expectancy and infant mortality rates among OECD nations and wealth inequality (as determined by the Gini Index).

Identify Patterns and Disparities: To find patterns of health outcome differences between nations with higher and lower income inequality levels in order to shed light on the ways in which wealth distribution affects public health.

Design and construct interactive data visualizations that enable people to investigate and comprehend the connections between socioeconomic variables and health outcomes.

Provide Perspectives for Policy Recommendations: to offer policymakers visual information that may aid in their understanding of how income inequality affects health outcomes, possibly directing future initiatives to lessen health disparities.

- **Which countries with high income inequality (Gini Index) have the lowest life expectancy?**
    - This will help users understand the relationship between income inequality and the general health of the population.
- **How does infant mortality vary across countries with different levels of income inequality?**
    - Users can explore how unequal income distribution impacts child health, which is a sensitive indicator of the overall health system.
- **What is the correlation between income inequality and life expectancy across OECD countries?**

    - Users will be able to see if there is a statistical relationship between income disparity and longevity in different countries.

Visualization Purpose

Illustrate the connection between health outcomes (life expectancy and infant mortality) and income disparity in OECD nations.

Focusing on Disparities: Give users the ability to visually contrast nations with varying degrees of income inequality and observe how these variations are represented in public health outcomes.

Make Data Accessible: This initiative intends to make the data more accessible to a wider audience, including researchers, policymakers, and the general public, by converting complicated datasets into understandable, user-friendly visuals.

Facilitate Exploration: Users can examine the data at their own pace, find trends, and derive insights that may impact conversations about social and healthcare issues by using interactive visualizations.

Data Processing

The data are cleaned with only basic adjustments to combine different indicators (life expectancy, income inequality and infant mortality) across countries and ensure that they are ready for analysis and the view. In the first step, all data sets were analyzed to ensure that only data from 2021 was used for analysis. Next, country names were checked for consistency across all datasets. Errors in country names were manually corrected to ensure accuracy. In addition, countries with missing data for all three indicators were excluded to maintain consistency in the analysis. This ensured that all countries had data on life expectancy, the Gini index and infant mortality.

Data Transformation

Data sets were combined into a single table using Excel. This allows copying the relevant columns (country, life expectancy, Gini index and infant mortality) onto a single page to create a combined data set for analysis.Rows with missing values were deleted as appropriate or added to average data from surrounding countries. This ensured that the final data set was free of gaps.

Country	Income Inequality(Gini Index)	Life expectancy(Number of Years	Infant Mortality Deaths Per 1000 live births
Costa Rica	0.487	84.5	9
Türkiye	0.433	83.6	8.7
Lithuania	0.366	83.3	4.9
Israel	0.348	83.2	4.3
Latvia	0.343	83.1	3.9
Japan	0.338	82.7	3.7
Italy	0.33	82.7	3.5
Korea	0.329	82.6	3.3

The Excel function =CORREL() was used to calculate the correlation between income inequality and life expectancy and income inequality and infant mortality. These correlations helped measure the strength of the relationship between income inequality and health outcomes.

Belgium	0.256	74.3	1.8
Czechia	0.255	74.2	1.7
Slovak Rep.	0.217	73.1	1.6
Correlation (Gini & Life Exp)		0.966824692	
Correlation (Gini & Infant Mortal		0.772795805	

Data Source

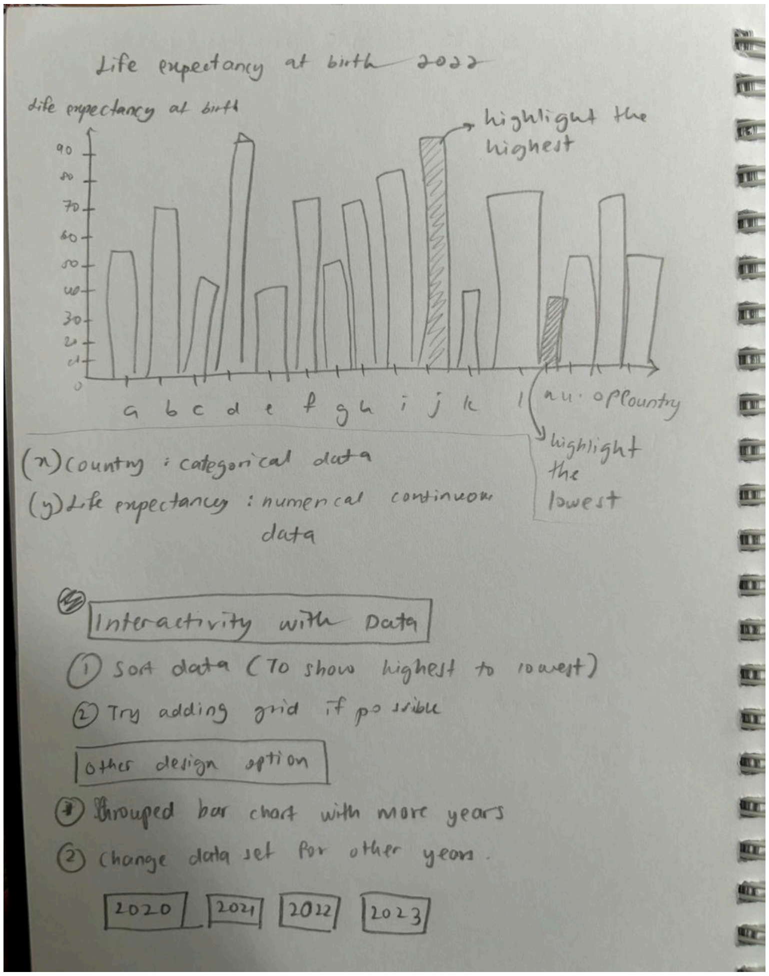
The data used for this visualization consists of three key health and socio-economic indicators, focusing on 2021 data for OECD and selected non-OECD countries. The data is tabular and includes the following attributes:

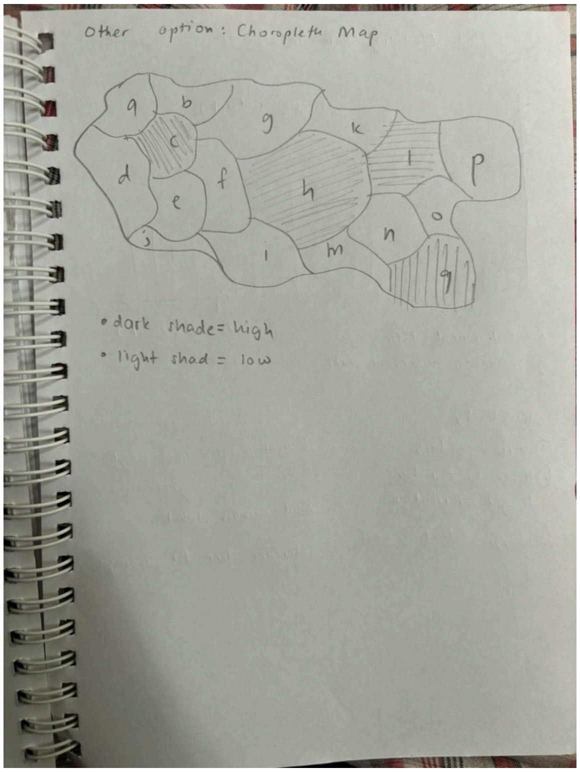
	Attribute	Description	Type	Country
Infant Mortality Rates	Deaths per 1,000 live births, 2021	This field contains the number of deaths per 1,000 live births in 2021. It is an indicator of the quality of healthcare and social systems within each	**Numerical (Continuous)**, measured on a scale between 0 and 1.	This field contains the name of the country for which the data is recorded.

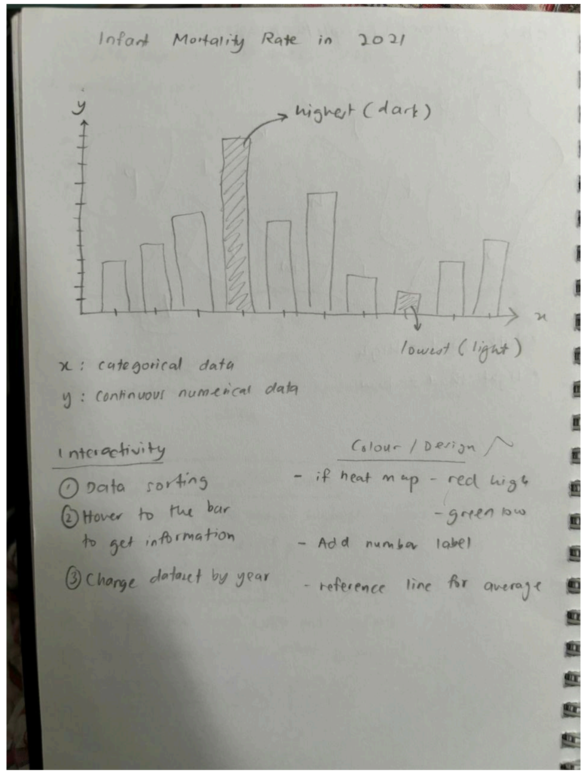
		country.		
Income Inequality (Gini Coefficient)	Gini Coefficient (0 = complete equality; 1 = complete inequality, 2021	This field contains the **Gini Index**, a measure of income inequality. It quantifies the income distribution of a country's residents on a scale from 0 (perfect equality) to 1 (maximum inequality).	**Numerical (Continuous)**, measured on a scale between 0 and 1	This field contains the name of the country for which the Gini Coefficient is recorded.
Life Expectancy at Birth	Total, Number of Years, 2021	This field contains the average number of years a newborn is expected to live if current mortality rates continue to apply. It is a general indicator of population health.	**Numerical (Continuous)**, measured in years	This field contains the name of the country for which life expectancy data is recorded.

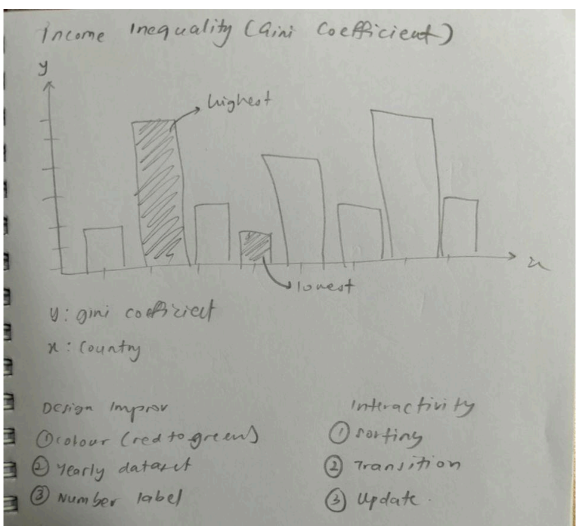
Initial Sketches and Ideas

For the initial sketch, we discussed what we would like to include for different visualisations. Bar chart seemed to be the most suitable chart for all the data, so it was included in the options. WE are yet to finalise the chart types for this one. The description of the chart is included in the image.

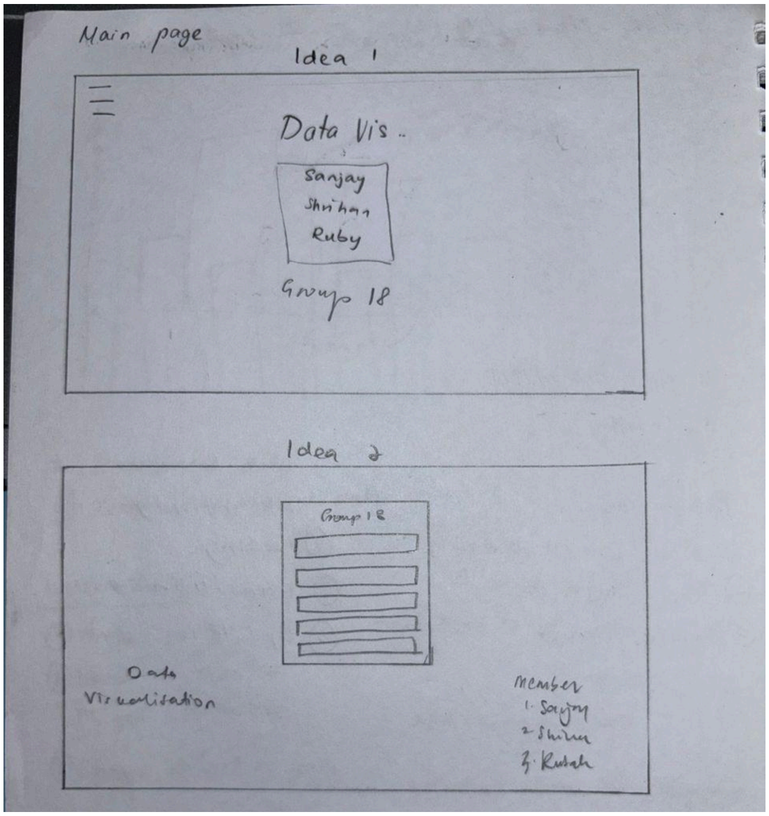



Other option: Choropleth Map
a b c d e f g h i j k l m n o p q
• dark shade = high
• light shad = low

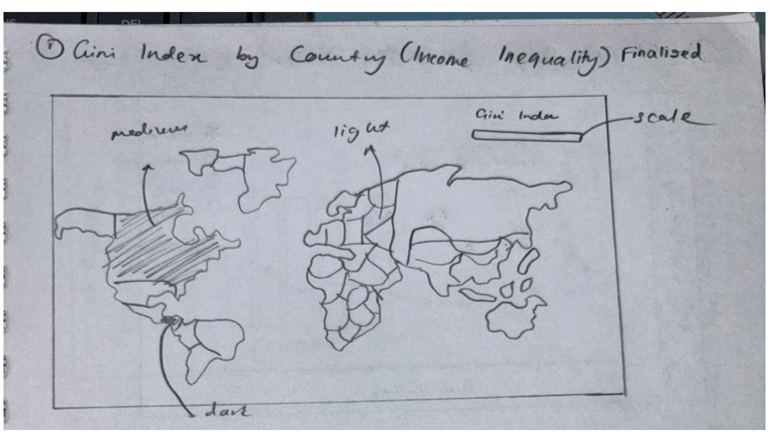

Infant Mortality Rate in 2021
y
x
highest (dark)
lowest (light)
x: categorical data
y: continuous numerical data
Interactivity
① Data sorting
② Hover to the bar to get information
③ Change dataset by year
Colour / Design
- if heat map - red high
- green low
- Add number label
- reference line for average


Income Inequality (Gini Coefficient)
y
x
highest
lowest
y: gini coefficient
x: country
Design Improv
① colour (red to green)
② Yearly dataset
③ Number label
Interactivity
① sorting
② Transition
③ Update

Finalised Sketch



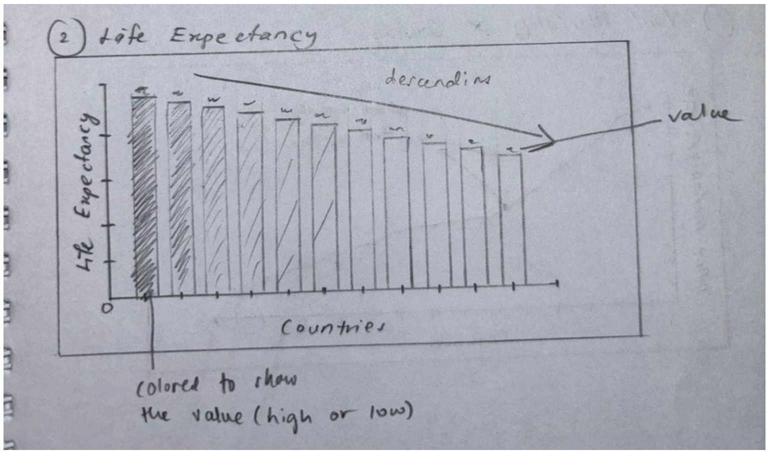
For the main page, we did sketches so we have a clearer image of how it would look like. We have the initial idea and the finalized idea.



1. Choropleth Map for Gini Index by Country (Income Inequality)

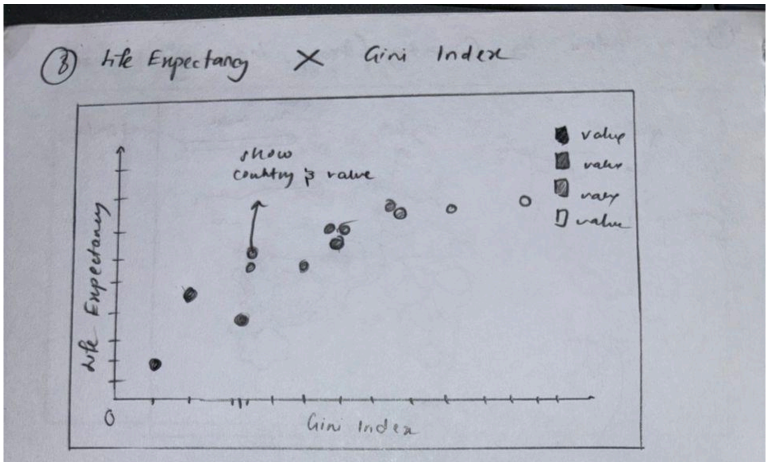
The first visualization is a choropleth map designed to depict income inequality across selected countries by visualizing the Gini Index. The map utilizes a color scale to indicate varying levels of income inequality, transitioning from lighter shades (representing lower Gini values, i.e., more income equality) to darker shades (indicating higher Gini values, i.e., more income inequality).

This approach allows for an intuitive comparison between countries. A color scale bar at the top right serves as a reference for interpreting the colors' meaning. When a user hovers over a specific country, detailed data, including the Gini Index value, will be displayed as a tooltip for enhanced interactivity and insight.



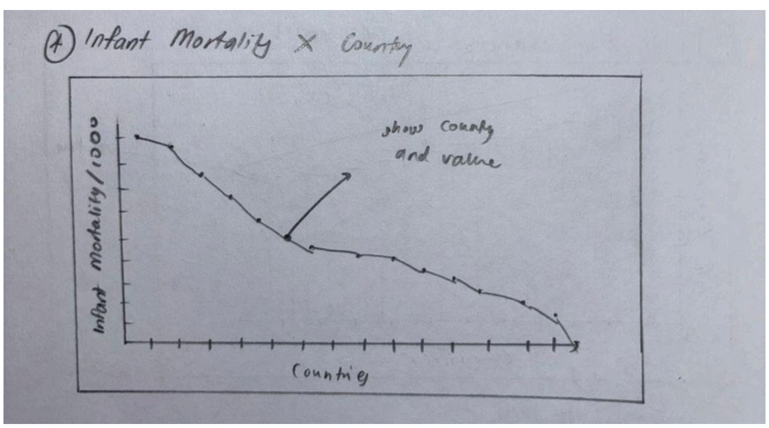
2. Bar Chart for Life Expectancy

The second sketch presents a bar chart showing life expectancy across different countries, arranged in descending order from highest to lowest. Each bar's color follows a gradient pattern where the tallest bar (representing the highest life expectancy) is the darkest, and the shortest bar (representing the lowest life expectancy) is the lightest. This gradient not only highlights the differences in life expectancy but also visually emphasizes the most and least performing countries. When a user hovers over a bar, a tooltip will display the exact data, and the value will also be shown at the top of each bar for clarity and immediate readability.



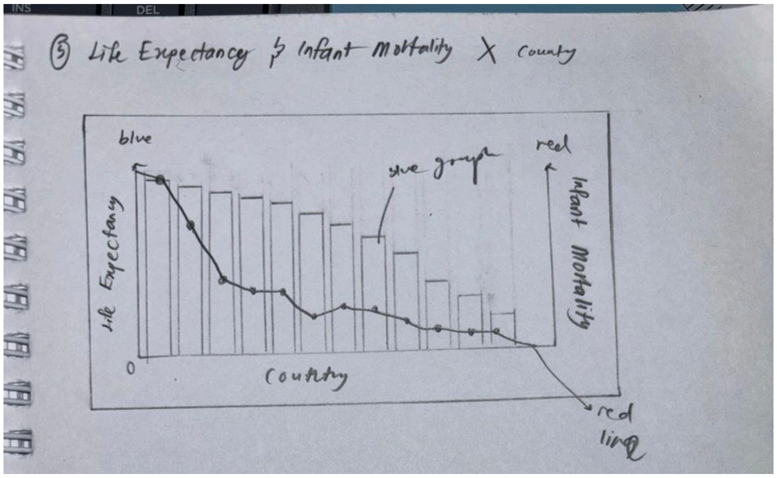
3. Scatter Plot: Life Expectancy vs. Gini Index

The third sketch features a scatter plot aimed at exploring the relationship between life expectancy and the Gini Index. Each circle on the plot represents a specific country, with its placement along the x-axis and y-axis corresponding to its Gini Index and life expectancy, respectively. Different colors are used to represent the countries, which helps distinguish data points. Hovering over a circle will display detailed information about the country, such as its name, Gini Index, and life expectancy value. This scatter plot effectively visualizes potential correlations or trends between income inequality and life expectancy.



4. Line Chart for Infant Mortality by Country

The fourth visualization is a line graph that shows the infant mortality rate for various countries, arranged in descending order. Each data point along the line represents a specific country and its corresponding infant mortality rate. The continuous line connecting the dots helps visualize the trend across the countries. When users hover over a data point, a tooltip will appear, displaying the country's name and its infant mortality rate, facilitating immediate access to specific data within the chart.



5. Combined Chart: Life Expectancy & Infant Mortality by Country

The fifth sketch combines a bar chart and a line chart to present a comprehensive comparison of life expectancy and infant mortality for the same set of countries. The x-axis represents the countries, while there are two separate y-axes: the left y-axis corresponds to life expectancy and is visualized using blue bars, while the right y-axis corresponds to infant mortality and is shown as a red line graph. This dual-axis chart provides a clear overview of how these two metrics relate and vary across the countries. Data points and bar values are highlighted with tooltips when hovered over, enabling users to access specific information effortlessly.

Initial Codes and Finalized Codes.

The following is the initial code for the website.

1. Javascript Files

JS script.js > createDualAxisChart > loadData() callback

```javascript
let dataLoaded = false;
let globalData = [];

// Load CSV data with exact column headers
function loadData(callback) {
    if (!dataLoaded) {
        d3.csv("Combined_Data.csv", d => ({
            Country: d["Country"],
            Gini_Index: +d["Gini_Index"],
            Life_Expectancy: +d["Life_Expectancy"],
            Infant_Mortality: +d["Infant_Mortality"]
        })).then(data => {
            globalData = data;
            dataLoaded = true;
            console.log("Data loaded:", globalData);  // Check data in the console
            callback();
        }).catch(error => {
            console.error("Error loading data:", error);
        });
    } else {
        callback();
    }
}

// Helper function for tooltips
function showTooltip(content, event) {
    d3.select(".tooltip")
        .style("left", (event.pageX + 10) + "px")
        .style("top", (event.pageY - 10) + "px")
        .style("display", "inline-block")
        .html(content);
}

function hideTooltip() {
    d3.select(".tooltip").style("display", "none");
}
```

```javascript
// Choropleth Map
function createChoroplethMap() {
  loadData(() => {
    const width = 1920, height = 650;
    const svg = d3.select("#mapChart").append("svg").attr("width", width).attr("height", height);
    const projection = d3.geoMercator().scale(140).translate([width / 2, height / 1.5]);
    const path = d3.geoPath().projection(projection);
    const colorScale = d3.scaleSequential(d3.interpolateYlGnBu).domain([0.2, 0.5]);

    // World map data
    d3.json("https://raw.githubusercontent.com/holtzy/D3-graph-gallery/master/DATA/world.geojson").then(world => {
      svg.selectAll("path")
          .data(world.features)
          .enter().append("path")
          .attr("d", path)
          .attr("fill", d => {
            const countryData = globalData.find(c => c.Country === d.properties.name);
            return countryData ? colorScale(countryData.Gini_Index) : "#e0e0e0";
          })
          .attr("stroke", "#555")
          .attr("stroke-width", 0.5)
          .on("mouseover", function() {
            d3.select(this).attr("stroke-width", 1.5).attr("stroke", "#333");
          })
          .on("mouseout", function() {
            d3.select(this).attr("stroke-width", 0.5).attr("stroke", "#555");
            hideTooltip();
          })
          .on("mousemove", function(event, d) {
            const countryData = globalData.find(c => c.Country === d.properties.name);
            if (countryData) {
              const tooltipContent = `<strong>Country:</strong> ${countryData.Country}<br>
                                      <strong>Gini Index:</strong> ${countryData.Gini_Index.toFixed(2)}<br>
                                      <strong>Life Expectancy:</strong> ${countryData.Life_Expectancy}<br>
                                      <strong>Infant Mortality:</strong> ${countryData.Infant_Mortality}`;
              showTooltip(tooltipContent, event);
            }
          });
    });
     //Legend
    const legendWidth = 200, legendHeight = 10;
    const legend = svg.append("g").attr("transform", `translate(${width - 400}, 60)`);
    // Gradient for legend
    const defs = svg.append("defs");
    const linearGradient = defs.append("linearGradient")
                               .attr("id", "legendGradient");
    linearGradient.selectAll("stop")
                  .data(colorScale.ticks(10).map((t, i, n) => ({ offset: `${100 * i / n.length}%`, color: colorScale(t) })))
                  .enter().append("stop")
                  .attr("offset", d => d.offset)
                  .attr("stop-color", d => d.color);
    legend.append("rect")
          .attr("width", legendWidth)
          .attr("height", legendHeight)
          .style("fill", "url(#legendGradient)")
          .style("stroke", "#ccc")
          .style("stroke-width", 0.5);
    // Legend Scale
    const legendScale = d3.scaleLinear().domain([0.2, 0.5]).range([0, legendWidth]);
    const legendAxis = d3.axisBottom(legendScale).ticks(5).tickFormat(d3.format(".2f"));
    legend.append("g").attr("transform", `translate(0,${legendHeight})`).call(legendAxis);
    legend.append("text")
          .attr("x", legendWidth / 2)
          .attr("y", -20)
          .attr("text-anchor", "middle")
          .style("font-size", "12px")
          .style("fill", "#333")
          .text("Gini Index");

  });
}
```

```
// Bar Chart
function createBarChart() {
  loadData(() => {
    const width = 1850, height = 700;
    const margin = { top: 50, right: 90, bottom: 180, left: 300 };
    const svg = d3.select("#barChart").append("svg")
                  .attr("width", width)
                  .attr("height", height);
    globalData.sort((a, b) => b.Life_Expectancy - a.Life_Expectancy);

    const x = d3.scaleBand()
                .domain(globalData.map(d => d.Country))
                .range([margin.left, width - margin.right])
                .padding(0.2);

    const y = d3.scaleLinear()
                .domain([0, d3.max(globalData, d => d.Life_Expectancy) + 5])
                .nice()
                .range([height - margin.bottom, margin.top]);

    const colorScale = d3.scaleSequential(d3.interpolateBlues)
                         .domain([0.2, 0.5]);

    // Draw bars
    svg.selectAll("rect")
      .data(globalData)
      .enter().append("rect")
      .attr("x", d => x(d.Country))
      .attr("y", d => y(d.Life_Expectancy))
      .attr("width", x.bandwidth())
      .attr("height", d => height - margin.bottom - y(d.Life_Expectancy))
      .attr("fill", d => colorScale(d.Gini_Index))
      .on("mousemove", (event, d) => {
        const tooltipContent = `<strong>Country:</strong> ${d.Country}<br>
                                <strong>Gini Index:</strong> ${d.Gini_Index}<br>
                                <strong>Life Expectancy:</strong> ${d.Life_Expectancy}<br>
                                <strong>Infant Mortality:</strong> ${d.Infant_Mortality}`;
        showTooltip(tooltipContent, event);
      })
      .on("mouseout", hideTooltip);

    // Add value labels above each bar showing Life Expectancy
    svg.selectAll(".label")
      .data(globalData)
      .enter().append("text")
      .attr("class", "label")
      .attr("x", d => x(d.Country) + x.bandwidth() / 2)
      .attr("y", d => y(d.Life_Expectancy) - 10)
      .attr("text-anchor", "middle")
      .style("font-size", "15px")
      .style("fill", "#333")
      .text(d => d.Life_Expectancy.toFixed(1));

    // Draw the x-axis at the bottom of the chart
    svg.append("g")
      .attr("transform", `translate(0,${height - margin.bottom})`)
      .call(d3.axisBottom(x).tickSizeOuter(0))
      .selectAll("text")
      .attr("transform", "rotate(-45)")
      .attr("dy", "0.75em")
      .attr("dx", "-0.75em")
      .style("text-anchor", "end")
      .style("font-size", "13px");

    // Draw the y-axis on the left of the chart
    svg.append("g")
      .attr("transform", `translate(${margin.left},0)`)
      .call(d3.axisLeft(y).ticks(5))
      .selectAll("text")
      .style("font-size", "10px");
```

```javascript
    // Add a label for the y-axis to indicate what the values represent
    svg.append("text")
        .attr("class", "axis-label")
        .attr("x", -height / 2.5)
        .attr("y", 250)
        .attr("transform", "rotate(-90)")
        .attr("text-anchor", "middle")
        .style("font-size", "20px")
        .style("fill", "#333")
        .text("Life Expectancy (Years)");

    // Add a label for the x-axis to indicate the countries represented
    svg.append("text")
        .attr("class", "axis-label")
        .attr("x", width / 2)
        .attr("y", height - 70)
        .attr("text-anchor", "middle")
        .style("font-size", "20px")
        .style("fill", "#333")
        .text("Country");
});
```

```javascript
function createScatterPlot() {
  loadData(() => {
    const width = 1700, height = 600, margin = { top: 50, right: 100, bottom: 70, left: 300 };
    const svg = d3.select("#scatterPlot").append("svg")
                  .attr("width", width)
                  .attr("height", height);

    const x = d3.scaleLinear().domain([0.2, 0.5]).range([margin.left, width - margin.right]);

    const y = d3.scaleLinear().domain([70, 90]).range([height - margin.bottom, margin.top]);

    const sizeScale = d3.scaleSqrt().domain([0, 10]).range([3, 15]);

    const colorScale = d3.scaleSequential(d3.interpolatePlasma)
                         .domain(d3.extent(globalData, d => d.Infant_Mortality));

    // Plotting circles
    svg.selectAll("circle")
      .data(globalData)
      .enter().append("circle")
      .attr("cx", d => x(d.Gini_Index))
      .attr("cy", d => y(d.Life_Expectancy))
      .attr("r", d => sizeScale(d.Infant_Mortality))
      .attr("fill", d => colorScale(d.Infant_Mortality))
      .attr("opacity", 0.85)

      .on("mousemove", (event, d) => {
        const tooltipContent = `<strong>Country:</strong> ${d.Country}<br>
                                <strong>Gini Index:</strong> ${d.Gini_Index}<br>
                                <strong>Life Expectancy:</strong> ${d.Life_Expectancy}<br>
                                <strong>Infant Mortality:</strong> ${d.Infant_Mortality}`;
        showTooltip(tooltipContent, event);
      })

      .on("mouseout", hideTooltip);

    // Draw x-axis
    svg.append("g")
      .attr("transform", `translate(0,${height - margin.bottom})`)
      .call(d3.axisBottom(x).tickFormat(d3.format(".2f")))
      .selectAll("text")
      .style("font-size", "13px");

    // X-axis label
    svg.append("text")
      .attr("x", width / 1.7)
      .attr("y", height - margin.bottom / 5.5)
      .attr("text-anchor", "middle")
      .style("font-size", "20px")
      .text("Gini Index");

    // Draw y-axis
    svg.append("g")
      .attr("transform", `translate(${margin.left},0)`)
      .call(d3.axisLeft(y))
      .selectAll("text")
      .style("font-size", "13px");

    // Y-axis label
    svg.append("text")
      .attr("transform", "rotate(-90)")
      .attr("y", margin.left / 1.1)
      .attr("x", -height / 2)
      .attr("dy", "-1.5em")
      .attr("text-anchor", "middle")
      .style("font-size", "20px")
      .text("Life Expectancy");
```

```
    // Legend for Infant Mortality with color scale
    const legend = svg.append("g")
                      .attr("transform", `translate(${width - margin.right - 30}, ${margin.top})`);

    // Draw color boxes for each tick in the color scale
    legend.selectAll("rect")
          .data(colorScale.ticks(5).slice(1))
          .enter().append("rect")
          .attr("y", (d, i) => i * 20)
          .attr("width", 15)
          .attr("height", 15)
          .style("fill", colorScale);

    // Add labels for each color box indicating Infant Mortality values
    legend.selectAll("text")
          .data(colorScale.ticks(5).slice(1))
          .enter().append("text")
          .attr("x", 20)
          .attr("y", (d, i) => i * 20 + 12)
          .text(d => d.toFixed(1))
          .style("font-size", "15px");
  });
}
```

```javascript
// Line Chart
function createLineChart() {
  loadData(() => {
    const width = 1570, height = 700, margin = { top: 60, right: 30, bottom: 180, left: 360 };
    const svg = d3.select("#lineChart").append("svg")
                  .attr("width", width)
                  .attr("height", height);

    const x = d3.scalePoint()
                .domain(globalData.map(d => d.Country))
                .range([margin.left, width - margin.right])
                .padding(0.5);

    // Define y-scale as a linear scale for Infant Mortality, setting range and limits
    const y = d3.scaleLinear()
                .domain([0, d3.max(globalData, d => d.Infant_Mortality) + 1])
                .nice()
                .range([height - margin.bottom, margin.top]);

    // Draw x-axis at the bottom of the chart with rotated text labels
    svg.append("g")
      .attr("transform", `translate(0,${height - margin.bottom})`)
      .call(d3.axisBottom(x))
      .selectAll("text")
      .attr("transform", "rotate(-45)")
      .style("text-anchor", "end")
      .style("font-size", "13px");

    // Draw y-axis on the left side of the chart
    svg.append("g")
      .attr("transform", `translate(${margin.left},0)`)
      .call(d3.axisLeft(y))
      .style("font-size", "15px");

    // Add horizontal gridlines for better readability of values
    svg.append("g")
      .attr("class", "grid")
      .attr("transform", `translate(${margin.left},0)`)
      .call(d3.axisLeft(y).ticks(5).tickSize(-width + margin.left + margin.right).tickFormat(""));

    // Define a line generator to create the line path based on country (x) and Infant Mortality (y)
    const line = d3.line()
                   .x(d => x(d.Country))
                   .y(d => y(d.Infant_Mortality));

    // Draw the line path on the SVG
    svg.append("path")
      .datum(globalData)
      .attr("fill", "none")
      .attr("stroke", "steelblue")
      .attr("stroke-width", 2)
      .attr("d", line);

    // Plot data points as circles on the line
    svg.selectAll("circle")
      .data(globalData)
      .enter().append("circle")
      .attr("cx", d => x(d.Country))
      .attr("cy", d => y(d.Infant_Mortality))
      .attr("r", 4)
      .attr("fill", "black")

      // Tooltip on mouse move
      .on("mousemove", (event, d) => {
        const tooltipContent = `<strong>Country:</strong> ${d.Country}<br>
                                <strong>Infant Mortality:</strong> ${d.Infant_Mortality}`;
        showTooltip(tooltipContent, event);
      })

      // Hide tooltip on mouse out
      .on("mouseout", hideTooltip);
```

```
    // Add x-axis label for country
    svg.append("text")
        .attr("x", width / 1.6)
        .attr("y", height - 70)
        .attr("text-anchor", "middle")
        .style("font-size", "20px")
        .text("Country");

    // Add y-axis label for Infant Mortality
    svg.append("text")
        .attr("x", -height / 2.5)
        .attr("y", 300)
        .attr("text-anchor", "middle")
        .attr("transform", "rotate(-90)")
        .style("font-size", "18px")
        .text("Infant Mortality (per 1000 live births)");
  });
}
```

```
// Dual-Axis Chart for Life Expectancy and Infant Mortality
function createDualAxisChart() {
 loadData(() => {
    const margin = { top: 50, right: 70, bottom: 180, left: 280 };
    const width = 1700 - margin.left - margin.right;
    const height = 700 - margin.top - margin.bottom;

    // Create SVG container and apply margins
    const svg = d3.select("#mapChart")
        .append("svg")
        .attr("width", width + margin.left + margin.right)
        .attr("height", height + margin.top + margin.bottom)
        .append("g")
        .attr("transform", `translate(${margin.left},${margin.top})`);

    // Axis Colors
    const leftAxisColor = "#1f77b4";
    const rightAxisColor = "#d62728";

    // X-axis
    const xScale = d3.scaleBand()
        .range([0, width])
        .padding(0.1);

    // Y-axes
    const yLeftScale = d3.scaleLinear().range([height, 0]);
    const yRightScale = d3.scaleLinear().range([height, 0]);

    // Load Data
    xScale.domain(globalData.map(d => d.Country));
    yLeftScale.domain([70, 90]);
    yRightScale.domain([0, 10]);

    // Add X-axis
    svg.append("g")
        .attr("transform", `translate(0,${height})`)
        .call(d3.axisBottom(xScale))
        .selectAll("text")
        .attr("text-anchor", "end")
        .attr("dx", "-0.8em")
        .attr("dy", "0.15em")
        .style("font-size", "15px")
        .attr("transform", "rotate(-45)");

    // Add Left Y-axis (Life Expectancy)
    svg.append("g")
        .style("color", leftAxisColor)
        .call(d3.axisLeft(yLeftScale))
        .append("text")
        .attr("transform", "rotate(-90)")
        .attr("x", -height / 2)
        .attr("y", -margin.left + 200)
        .attr("dy", "1em")
        .style("text-anchor", "middle")
        .style("fill", leftAxisColor)
        .style("font-size", "20px")
        .text("Life Expectancy (Years)");

    // Add Right Y-axis (Infant Mortality)
    svg.append("g")
        .attr("transform", `translate(${width},0)`)
        .style("color", rightAxisColor)
        .call(d3.axisRight(yRightScale))
        .append("text")
        .attr("transform", "rotate(-90)")
        .attr("x", -height / 2)
        .attr("y", margin.right - 20)
        .attr("dy", "1em")
        .style("text-anchor", "middle")
        .style("fill", rightAxisColor)
        .style("font-size", "18px")
        .text("Infant Mortality (per 1000 live births)");
```

```
    // Bars for Life Expectancy
    svg.selectAll(".bar")
        .data(globalData)
        .enter().append("rect")
        .attr("class", "bar")
        .attr("x", d => xScale(d.Country))
        .attr("y", d => yLeftScale(d.Life_Expectancy))
        .attr("width", xScale.bandwidth())
        .attr("height", d => height - yLeftScale(d.Life_Expectancy))
        .style("fill", leftAxisColor);

    // Line and Points for Infant Mortality
    const line = d3.line()
        .x(d => xScale(d.Country) + xScale.bandwidth() / 2)
        .y(d => yRightScale(d.Infant_Mortality));

    svg.append("path")
        .datum(globalData)
        .attr("fill", "none")
        .attr("stroke", rightAxisColor)
        .attr("stroke-width", 1.5)
        .attr("d", line);

    svg.selectAll(".point")
        .data(globalData)
        .enter().append("circle")
        .attr("class", "point")
        .attr("cx", d => xScale(d.Country) + xScale.bandwidth() / 2)
        .attr("cy", d => yRightScale(d.Infant_Mortality))
        .attr("r", 4)
        .style("fill", rightAxisColor);

    // Tooltip for Data Points
    const tooltip = d3.select("body").append("div")
        .attr("class", "tooltip")
        .style("opacity", 0);

    svg.selectAll(".point")
        .on("mouseover", function(event, d) {
            tooltip.transition().duration(200).style("opacity", .9);
            tooltip.html(`<strong>Country:</strong> ${d.Country}<br>
                          <strong>Life Expectancy:</strong> ${d.Life_Expectancy}<br>
                          <strong>Infant Mortality:</strong> ${d.Infant_Mortality}`)
                .style("left", (event.pageX + 5) + "px")
                .style("top", (event.pageY - 28) + "px");
        })
        .on("mouseout", function() {
            tooltip.transition().duration(500).style("opacity", 0);
        });
});
```

2. Index HTML

index.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Data Visualizations</title>
  <link rel="stylesheet" href="style.css">
</head>
<body class="index-background">
  <div class="container">
    <h1>GROUP 18</h1>
    <p>Comparison between data's with different Visualization.</p>
    <div class="link-container">
      <a href="map_chart.html">Map Chart</a>
      <a href="bar_chart.html">Bar Chart</a>
      <a href="scatter_plot.html">Scatter Plot</a>
      <a href="line_chart.html">Line Chart</a>
      <a href="dual_axis_chart.html">Dual-Axis Chart</a>
    </div>
    <div class="footer">TOGETHER, WE CAN MAKE A DIFFERENCE.</div>
  </div>
</body>
</html>

```

3. Map Chart

map_chart.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Choropleth Map</title>
  <link rel="stylesheet" href="style.css">
  <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
  <div class="title-container">
    <h2>Choropleth Map: Gini Index by Country (Income Inequality)</h2>
    <p>Choropleth map showing income inequality across different countries</p>
  </div>
  <div id="mapChart" class="chart"></div>
  <div class="tooltip"></div>
  <script src="script.js"></script>
  <script>createChoroplethMap();</script>
</body>
</html>

```

4. Bar Chart

bar_chart.html > html > body > p

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Bar Chart</title>
  <link rel="stylesheet" href="style.css">
  <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
  <h2>Bar Chart: Life Expectancy with Gini Index and Infant Mortality</h2>
  <p>Sorted by Life Expectancy for Enhanced Comparison</p>
  <div id="barChart" class="chart"></div>
  <div class="tooltip"></div>
  <script src="script.js"></script>
  <script>createBarChart();</script>
</body>
</html>
```

5. Dual-axis Chart

dual_axis_chart.html > html > body > p

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Dual-Axis Chart</title>
  <link rel="stylesheet" href="style.css">
  <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
  <h2>Dual-Axis Chart: Life Expectancy & Infant Mortality by Country</h2>
  <p>Comparing Life Expectancy and Infant Mortality across countries</p>
  <div id="mapChart" class="chart"></div>
  <div class="tooltip"></div>
  <script src="script.js"></script>
  <script>createDualAxisChart();</script>
</body>
</html>
```

6. Scatter Plot Chart

```
scatter_plot.html > html > body > p
<html lang="en">
<head>
    <title>Scatter Plot</title>
    <link rel="stylesheet" href="style.css">
    <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
    <h2>Scatter Plot: Life Expectancy vs Gini Index with Infant Mortality</h2>
    <p>Impact of Income Inequality on Life Expectancy with Infant Mortality</p>
    <div id="scatterPlot" class="chart"></div>
    <div class="tooltip"></div>
    <script src="script.js"></script>
    <script>createScatterPlot();</script>
</body>
</html>
```

The following is the finalized code for the website.

```
map_chart.html > ...
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Choropleth Map</title>
    <link rel="stylesheet" href="style.css">
    <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
    <h2>Choropleth Map: Gini Index by Country</h2>
    <div id="mapChart" class="chart"></div>
    <div class="tooltip"></div>
    <script src="script.js"></script>
    <script>createChoroplethMap();</script>
</body>
</html>
```

Figure 1. Map_chart.html

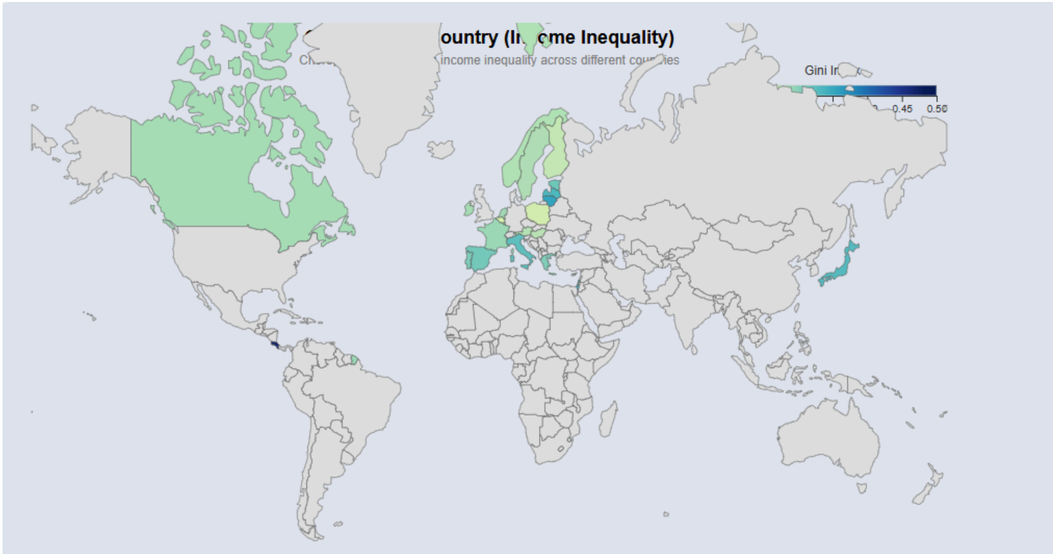


Figure 2. Choropleth Map Visualization

Choropleth Map

The map visualization is a choropleth representation displaying the Gini Index in different nations, reflecting income disparity. It also includes information about infant mortality and life expectancy, although these specifics are not visually distinguishable on the map itself. Every nation is colored according to its Gini Index value, with shades from light to dark representing lower to higher income inequality, respectively. This map is designed to offer a worldwide comparison of income disparity levels, enabling users to swiftly recognize areas with greater or lesser inequality.

```javascript
// Choropleth Map
function createChoroplethMap() {
  loadData(() => {
    const width = 900, height = 500;
    const svg = d3.select("#mapChart").append("svg").attr("width", width).attr("height", height);
    const projection = d3.geoMercator().scale(140).translate([width / 2, height / 1.5]);
    const path = d3.geoPath().projection(projection);
    // Updated color scale for better color contrast
    const colorScale = d3.scaleSequential(d3.interpolateYlGnBu).domain([0.2, 0.5]);
    // Title and Subtitle
    svg.append("text")
        .attr("x", width / 2)
        .attr("y", 20)
        .attr("text-anchor", "middle")
        .style("font-size", "18px")
        .style("font-weight", "bold")
        .text("Gini Index by Country (Income Inequality)");
    svg.append("text")
        .attr("x", width / 2)
        .attr("y", 40)
        .attr("text-anchor", "middle")
        .style("font-size", "12px")
        .style("fill", "gray")
        .text("Choropleth map showing income inequality across different countries");
    // World map data
    d3.json("https://raw.githubusercontent.com/holtzy/D3-graph-gallery/master/DATA/world.geojson").then(world => {
      svg.selectAll("path")
        .data(world.features)
        .enter().append("path")
        .attr("d", path)
        .attr("fill", d => {
          const countryData = globalData.find(c => c.Country === d.properties.name);
          return countryData ? colorScale(countryData.Gini_Index) : "#e0e0e0";
        })
        .attr("stroke", "#555")
        .attr("stroke-width", 0.5)
        .on("mouseover", function() {
          d3.select(this).attr("stroke-width", 1.5).attr("stroke", "#333");
        })
        .on("mouseout", function() {
          d3.select(this).attr("stroke-width", 0.5).attr("stroke", "#555");
          hideTooltip();
        })
        .on("mousemove", function(event, d) {
          const countryData = globalData.find(c => c.Country === d.properties.name);
          if (countryData) {
            const tooltipContent = `<strong>Country:</strong> ${countryData.Country}<br>
                                    <strong>Gini Index:</strong> ${countryData.Gini_Index.toFixed(2)}<br>
                                    <strong>Life Expectancy:</strong> ${countryData.Life_Expectancy}<br>
                                    <strong>Infant Mortality:</strong> ${countryData.Infant_Mortality}`;
            showTooltip(tooltipContent, event);
          }
        });
    });
    // Improved Legend
    const legendWidth = 200, legendHeight = 10;
    const legend = svg.append("g").attr("transform", `translate(${width - 220}, 60)`);
    // Gradient for legend
    const defs = svg.append("defs");
    const linearGradient = defs.append("linearGradient")
                               .attr("id", "legendGradient");
    linearGradient.selectAll("stop")
                  .data(colorScale.ticks(10).map((t, i, n) => ({ offset: `${100 * i / n.length}%`, color: colorScale(t) })))
                  .enter().append("stop")
                  .attr("offset", d => d.offset)
                  .attr("stop-color", d => d.color);
    legend.append("rect")
          .attr("width", legendWidth)
          .attr("height", legendHeight)
          .style("fill", "url(#legendGradient)")
          .style("stroke", "#ccc")
          .style("stroke-width", 0.5);
    // Legend Scale
    const legendScale = d3.scaleLinear().domain([0.2, 0.5]).range([0, legendWidth]);
    const legendAxis = d3.axisBottom(legendScale).ticks(5).tickFormat(d3.format(".2f"));
    legend.append("g").attr("transform", `translate(0,${legendHeight})`).call(legendAxis);
    legend.append("text")
          .attr("x", legendWidth / 2)
          .attr("y", -10)
          .attr("text-anchor", "middle")
          .style("font-size", "12px")
          .style("fill", "#333")
          .text("Gini Index");
  });
}
```

Figure 3. Javascript for the map_chart.

Suggested Enhancement

To enhance this visualization, think about incorporating unique markers or color-coded overlays for infant mortality, allowing users to quickly observe both income inequality and infant mortality simultaneously. Furthermore, moving the cursor over each nation might activate a tooltip that shows detailed figures for the Gini Index, infant mortality rate, and life expectancy. Improving the color contrast on the Gini Index scale and making sure that labels and titles are distinct and do not overlap with map elements would boost readability and engagement.

```
bar_chart.html > html > head > link
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Bar Chart</title>
  <link rel="stylesheet" href="style.css">
  <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
  <h2>Bar Chart: Life Expectancy with Gini Index and Infant Mortality</h2>
  <div id="barChart" class="chart"></div>
  <div class="tooltip"></div>
  <script src="script.js"></script>
  <script>createBarChart();</script>
</body>
</html>
```

Figure 4. bar_chart.html

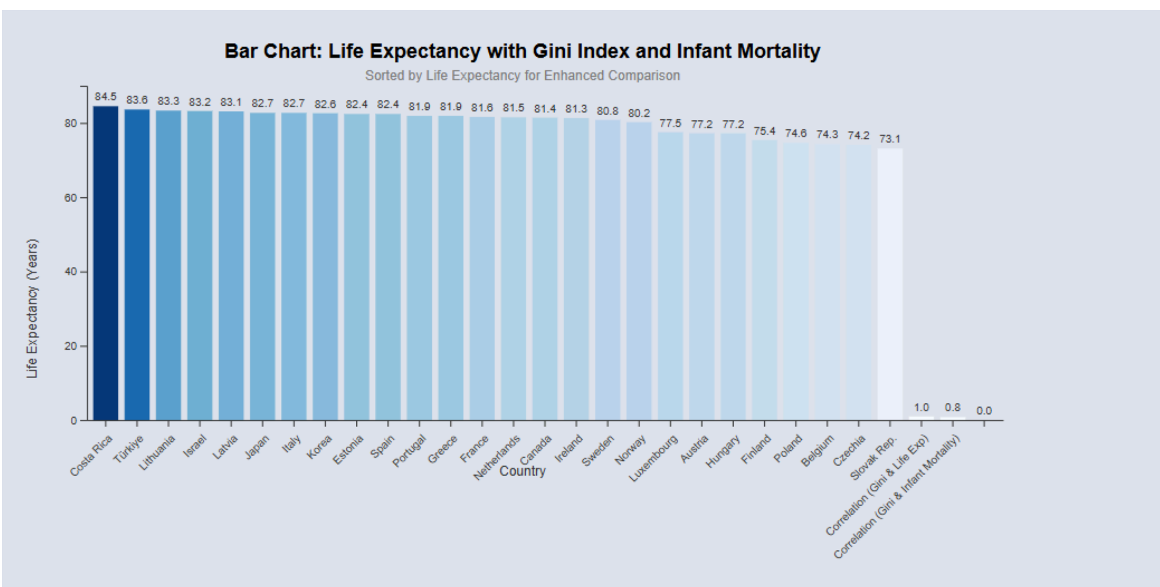


Figure 5. Bar Chart Visualization

Sketch Description: This bar graph depicts the life expectancy across different nations, employing color intensity to indicate the Gini Index for each country. The chart is arranged according to life expectancy, facilitating straightforward comparisons between countries with longer and shorter lifespans. Moreover, infant mortality data points are integrated as labels on each bar, illustrating a complex relationship between income inequality and health metrics.

```
function createBarChart() {
  loadData(() => {
    const width = 900, height = 500;
    const margin = { top: 50, right: 20, bottom: 150, left: 60 };

    const svg = d3.select("#barChart").append("svg")
                  .attr("width", width)
                  .attr("height", height);

    // Sort data by Life Expectancy for better readability
    globalData.sort((a, b) => b.Life_Expectancy - a.Life_Expectancy);

    // Scales
    const x = d3.scaleBand()
                .domain(globalData.map(d => d.Country))
                .range([margin.left, width - margin.right])
                .padding(0.2);

    const y = d3.scaleLinear()
                .domain([0, d3.max(globalData, d => d.Life_Expectancy) + 5])
                .nice()
                .range([height - margin.bottom, margin.top]);

    const colorScale = d3.scaleSequential(d3.interpolateBlues)
                         .domain([0.2, 0.5]);

    // Bars
    svg.selectAll("rect")
      .data(globalData)
      .enter().append("rect")
      .attr("x", d => x(d.Country))
      .attr("y", d => y(d.Life_Expectancy))
      .attr("width", x.bandwidth())
      .attr("height", d => height - margin.bottom - y(d.Life_Expectancy))
      .attr("fill", d => colorScale(d.Gini_Index))
      .on("mousemove", (event, d) => {
        const tooltipContent = `<strong>Country:</strong> ${d.Country}<br>
                                <strong>Gini Index:</strong> ${d.Gini_Index}<br>
                                <strong>Life Expectancy:</strong> ${d.Life_Expectancy}<br>
                                <strong>Infant Mortality:</strong> ${d.Infant_Mortality}`;
        showTooltip(tooltipContent, event);
      })
      .on("mouseout", hideTooltip);

    // Value Labels on top of bars
    svg.selectAll(".label")
      .data(globalData)
      .enter().append("text")
      .attr("class", "label")
      .attr("x", d => x(d.Country) + x.bandwidth() / 2)
      .attr("y", d => y(d.Life_Expectancy) - 5)
      .attr("text-anchor", "middle")
      .style("font-size", "10px")
      .style("fill", "#333")
      .text(d => d.Life_Expectancy.toFixed(1));

    // X Axis
    svg.append("g")
      .attr("transform", `translate(0,${height - margin.bottom})`)
      .call(d3.axisBottom(x).tickSizeOuter(0))
      .selectAll("text")
      .attr("transform", "rotate(-45)")
      .attr("dy", "0.75em")
      .attr("dx", "-0.75em")
      .style("text-anchor", "end")
      .style("font-size", "10px");

    // Y Axis
    svg.append("g")
      .attr("transform", `translate(${margin.left},0)`)
      .call(d3.axisLeft(y).ticks(5))
      .selectAll("text")
      .style("font-size", "10px");
```

```javascript
// Y Axis Label
svg.append("text")
    .attr("class", "axis-label")
    .attr("x", -height / 2)
    .attr("y", 15)
    .attr("transform", "rotate(-90)")
    .attr("text-anchor", "middle")
    .style("font-size", "12px")
    .style("fill", "#333")
    .text("Life Expectancy (Years)");

// X Axis Label
svg.append("text")
    .attr("class", "axis-label")
    .attr("x", width / 2)
    .attr("y", height - 100)
    .attr("text-anchor", "middle")
    .style("font-size", "12px")
    .style("fill", "#333")
    .text("Country");

// Title
svg.append("text")
    .attr("x", width / 2)
    .attr("y", 25)
    .attr("text-anchor", "middle")
    .style("font-size", "18px")
    .style("font-weight", "bold")
    .text("Bar Chart: Life Expectancy with Gini Index and Infant Mortality");

svg.append("text")
    .attr("x", width / 2)
    .attr("y", 45)
    .attr("text-anchor", "middle")
    .style("font-size", "12px")
    .style("fill", "gray")
    .text("Sorted by Life Expectancy for Enhanced Comparison");
);
```

Figure 6. Javascript for Bar Chart

Improvement Recommendations: To enhance understanding, think about including a legend for the Gini Index shading, which will facilitate the interpretation of changes in income inequality. Additionally, positioning data labels directly above each bar might enhance readability for life expectancy and infant mortality figures, allowing viewers to easily understand the comparison without needing to follow values along the axis.

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Scatter Plot</title>
  <link rel="stylesheet" href="style.css">
  <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
  <h2>Scatter Plot: Life Expectancy vs Gini Index with Infant Mortality</h2>
  <div id="scatterPlot" class="chart"></div>
  <div class="tooltip"></div>
  <script src="script.js"></script>
  <script>createScatterPlot();</script>
</body>
</html>
```

Figure 7. Scatter_plot.html

Sketch Description: This scatter plot illustrates the connection between the Gini Index (income inequality) and life expectancy in various countries, where the size and color of each point indicate infant mortality rates. Nations exhibiting lower Gini Index figures and greater life expectancy group towards the left and upper part of the graph, suggesting a possible link between reduced income disparity and extended life durations. Bigger, more vividly colored dots represent greater infant mortality, offering an extra dimension of understanding regarding health inequalities.

```javascript
function createScatterPlot() {
  loadData(() => {
    const width = 800, height = 500, margin = { top: 50, right: 20, bottom: 70, left: 70 };

    const svg = d3.select("#scatterPlot").append("svg")
                  .attr("width", width)
                  .attr("height", height);

    const x = d3.scaleLinear().domain([0.2, 0.5]).range([margin.left, width - margin.right]);
    const y = d3.scaleLinear().domain([70, 90]).range([height - margin.bottom, margin.top]);
    const sizeScale = d3.scaleSqrt().domain([0, 10]).range([3, 15]);

    // Adjusted Color scale for Infant Mortality (starting with a darker shade)
    const colorScale = d3.scaleSequential(d3.interpolatePlasma) // Plasma has good contrast in lighter colors
                          .domain(d3.extent(globalData, d => d.Infant_Mortality));

    // Title for the plot
    svg.append("text")
      .attr("x", width / 2)
      .attr("y", margin.top / 2)
      .attr("text-anchor", "middle")
      .attr("class", "title")
      .style("font-size", "16px")
      .style("fill", "#663399")  // Dark purple for title
      .text("Impact of Income Inequality on Life Expectancy with Infant Mortality");

    // Plotting circles with improved opacity
    svg.selectAll("circle")
      .data(globalData)
      .enter().append("circle")
      .attr("cx", d => x(d.Gini_Index))
      .attr("cy", d => y(d.Life_Expectancy))
      .attr("r", d => sizeScale(d.Infant_Mortality))
      .attr("fill", d => colorScale(d.Infant_Mortality))
      .attr("opacity", 0.85)  // Increased opacity for better visibility
      .on("mousemove", (event, d) => {
        const tooltipContent = `<strong>Country:</strong> ${d.Country}<br>
                                <strong>Gini Index:</strong> ${d.Gini_Index}<br>
                                <strong>Life Expectancy:</strong> ${d.Life_Expectancy}<br>
                                <strong>Infant Mortality:</strong> ${d.Infant_Mortality}`;
        showTooltip(tooltipContent, event);
      })
      .on("mouseout", hideTooltip);

    // X-axis
    svg.append("g")
      .attr("transform", `translate(0,${height - margin.bottom})`)
      .call(d3.axisBottom(x).tickFormat(d3.format(".2f")));

    // X-axis label
    svg.append("text")
      .attr("x", width / 2)
      .attr("y", height - margin.bottom / 3)
      .attr("text-anchor", "middle")
      .style("font-size", "12px")
      .text("Gini Index");

    // Y-axis
    svg.append("g")
      .attr("transform", `translate(${margin.left},0)`)
      .call(d3.axisLeft(y));

    // Y-axis label
    svg.append("text")
      .attr("transform", "rotate(-90)")
      .attr("y", margin.left / 3)
      .attr("x", -height / 2)
      .attr("dy", "-1.5em")
      .attr("text-anchor", "middle")
      .style("font-size", "12px")
      .text("Life Expectancy");

    // Legend for Infant Mortality with adjusted colors
    const legend = svg.append("g")
                      .attr("transform", `translate(${width - margin.right - 30}, ${margin.top})`);

    legend.selectAll("rect")
        .data(colorScale.ticks(5).slice(1))
        .enter().append("rect")
        .attr("y", (d, i) => i * 20)
        .attr("width", 15)
        .attr("height", 15)
        .style("fill", colorScale);

    legend.selectAll("text")
        .data(colorScale.ticks(5).slice(1))
        .enter().append("text")
        .attr("x", 20)
        .attr("y", (d, i) => i * 20 + 12)
        .text(d => d.toFixed(1))
        .style("font-size", "10px");
  });
}
```

Figure 8. Javascript for Scatter Plot Visualization

Improvement Recommendations: To enhance clarity, a more noticeable legend that explains the color gradient and point sizes related to infant mortality could be introduced. Incorporating gridlines or reference lines for the average values of the Gini Index and life expectancy could assist viewers in easily spotting outliers and overall trends, enhancing the chart's insightfulness.

line_chart.html > html > head > link

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Line Chart</title>
    <link rel="stylesheet" href="style.css">
    <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
    <h2>Line Chart: Infant Mortality by Country</h2>
    <div id="lineChart" class="chart"></div>
    <div class="tooltip"></div>
    <script src="script.js"></script>
    <script>createLineChart();</script>
</body>
</html>
```

Figure 9. Line_chart.html

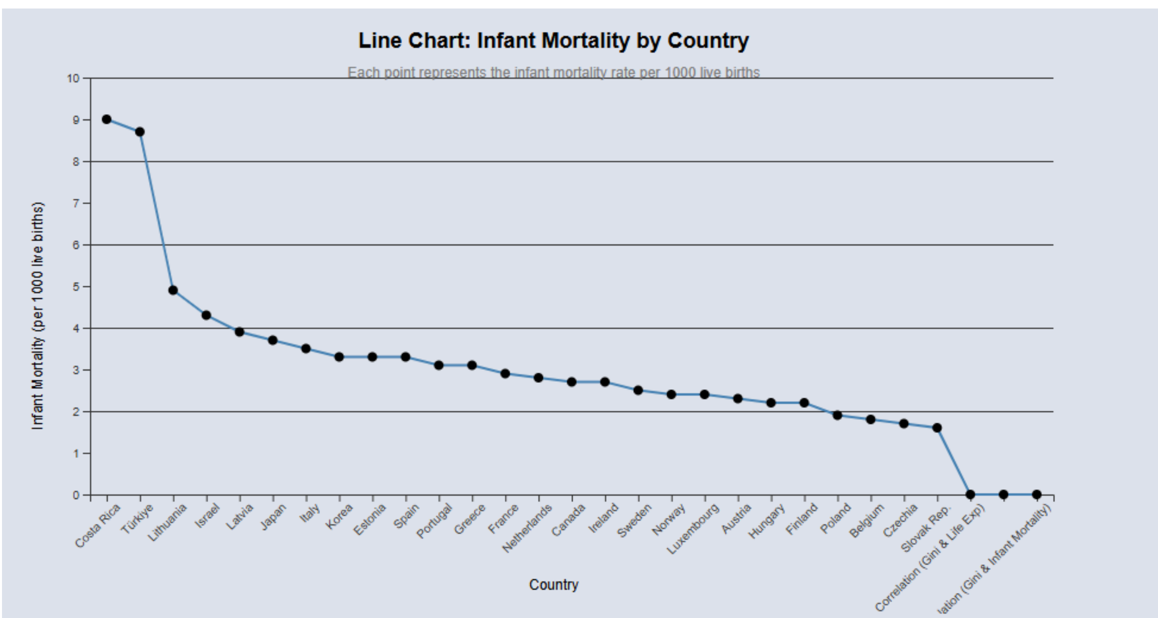


Figure 10. Line Chart Visualization

Sketch Description: This line graph illustrates infant death rates in different countries, with each point indicating the number of infant fatalities per 1,000 live births. Nations are placed along the x-axis, organized by their infant mortality rate in descending order from left to right. The y-axis displays infant mortality rates, varying from zero to ten, facilitating a straightforward visual comparison among nations.

```
function createLineChart() {
  loadData(() => {
    const width = 900, height = 500, margin = { top: 50, right: 30, bottom: 100, left: 60 };
    const svg = d3.select("#lineChart").append("svg")
                  .attr("width", width)
                  .attr("height", height);

    // Scales
    const x = d3.scalePoint()
                .domain(globalData.map(d => d.Country))
                .range([margin.left, width - margin.right])
                .padding(0.5);

    const y = d3.scaleLinear()
                .domain([0, d3.max(globalData, d => d.Infant_Mortality) + 1])
                .nice()
                .range([height - margin.bottom, margin.top]);

    // Axes
    svg.append("g")
       .attr("transform", `translate(0,${height - margin.bottom})`)
       .call(d3.axisBottom(x))
       .selectAll("text")
       .attr("transform", "rotate(-45)")
       .style("text-anchor", "end")
       .style("font-size", "10px");

    svg.append("g")
       .attr("transform", `translate(${margin.left},0)`)
       .call(d3.axisLeft(y));

    // Gridlines
    svg.append("g")
       .attr("class", "grid")
       .attr("transform", `translate(${margin.left},0)`)
       .call(d3.axisLeft(y).ticks(5).tickSize(-width + margin.left + margin.right).tickFormat(""));

    // Line
    const line = d3.line()
                   .x(d => x(d.Country))
                   .y(d => y(d.Infant_Mortality));

    svg.append("path")
       .datum(globalData)
       .attr("fill", "none")
       .attr("stroke", "steelblue")
       .attr("stroke-width", 2)
       .attr("d", line);

    // Data points
    svg.selectAll("circle")
       .data(globalData)
       .enter().append("circle")
       .attr("cx", d => x(d.Country))
       .attr("cy", d => y(d.Infant_Mortality))
       .attr("r", 4)
       .attr("fill", "black")
       .on("mousemove", (event, d) => {
         const tooltipContent = `<strong>Country:</strong> ${d.Country}<br>
                                 <strong>Infant Mortality:</strong> ${d.Infant_Mortality}`;
         showTooltip(tooltipContent, event);
       })
       .on("mouseout", hideTooltip);

    // Title and Subtitle
    svg.append("text")
       .attr("x", width / 2)
       .attr("y", margin.top / 2)
       .attr("text-anchor", "middle")
       .style("font-size", "18px")
       .style("font-weight", "bold")
       .text("Line Chart: Infant Mortality by Country");

    svg.append("text")
       .attr("x", width / 2)
       .attr("y", margin.top)
       .attr("text-anchor", "middle")
       .style("font-size", "12px")
       .style("fill", "gray")
       .text("Each point represents the infant mortality rate per 1000 live births");

    // Axis Labels
    svg.append("text")
       .attr("x", width / 2)
       .attr("y", height - 20)
       .attr("text-anchor", "middle")
       .style("font-size", "12px")
       .text("Country");

    svg.append("text")
       .attr("x", -height / 2)
       .attr("y", 20)
       .attr("text-anchor", "middle")
       .attr("transform", "rotate(-90)")
       .style("font-size", "12px")
       .text("Infant Mortality (per 1000 live births)");
  });
}
```

Improvement Ideas: Implementing color coding to emphasize nations with notably high or low infant mortality may help highlight significant differences more effectively. Moreover, notes at specific locations (such as the peak and trough mortality rates) might enhance the emphasis on important data points and increase the narrative value of the chart. Incorporating labels for average or standard mortality rates could help in putting these rates into context.

dual_axis_chart.html > html > head > link

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Dual-Axis Chart</title>
    <link rel="stylesheet" href="style.css">
    <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
    <h2>Dual-Axis Chart: Life Expectancy & Infant Mortality by Country</h2>
    <div id="dualAxisChart" class="chart"></div>
    <div class="tooltip"></div>
    <script src="script.js"></script>
    <script>createDualAxisChart();</script>
</body>
</html>
```

Figure 12. Dual_axis_chart.html

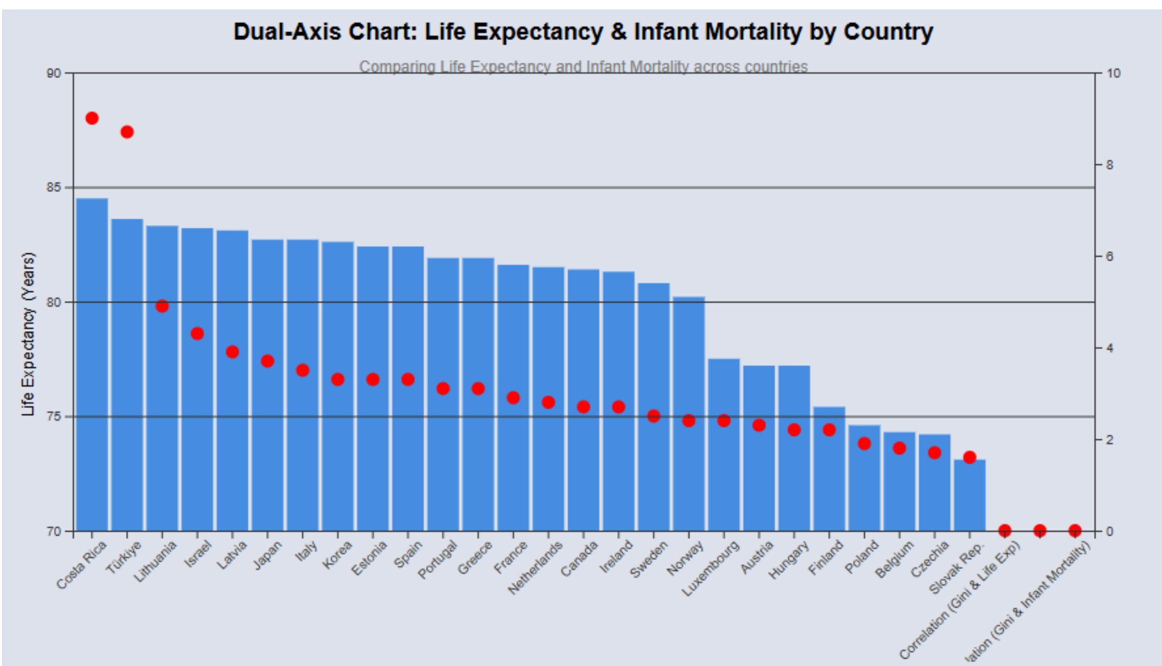


Figure 12. Dual_axis_chart.html

Sketch Description: This two-dimensional chart illustrates Life Expectancy and Infant Mortality statistics for various countries. The blue bars indicate life expectancy in years on the left y-axis, whereas the red dots illustrate infant mortality rates for every 1,000 live births on the right y-axis. The x-axis arranges countries in descending order of life expectancy for clearer comparison. This arrangement enables a fast, side-by-side evaluation of the two metrics among countries.

```javascript
function createDualAxisChart() {
  loadData(() => {
    const width = 900, height = 500, margin = { top: 50, right: 60, bottom: 100, left: 60 };
    const svg = d3.select("#dualAxisChart").append("svg")
                  .attr("width", width)
                  .attr("height", height);

    // Scales
    const x = d3.scaleBand()
                .domain(globalData.map(d => d.Country))
                .range([margin.left, width - margin.right])
                .padding(0.1);

    const yLeft = d3.scaleLinear()
                    .domain([70, d3.max(globalData, d => d.Life_Expectancy) + 5])
                    .nice()
                    .range([height - margin.bottom, margin.top]);

    const yRight = d3.scaleLinear()
                     .domain([0, d3.max(globalData, d => d.Infant_Mortality) + 1])
                     .nice()
                     .range([height - margin.bottom, margin.top]);

    // Bars for Life Expectancy
    svg.selectAll(".bar")
      .data(globalData)
      .enter().append("rect")
      .attr("x", d => x(d.Country))
      .attr("y", d => yLeft(d.Life_Expectancy))
      .attr("width", x.bandwidth())
      .attr("height", d => height - margin.bottom - yLeft(d.Life_Expectancy))
      .attr("fill", "#4a90e2")
      .on("mousemove", (event, d) => {
        const tooltipContent = `<strong>Country:</strong> ${d.Country}<br>
                                <strong>Life Expectancy:</strong> ${d.Life_Expectancy}`;
        showTooltip(tooltipContent, event);
      })
      .on("mouseout", hideTooltip);

    // Points for Infant Mortality
    svg.selectAll(".circle")
      .data(globalData)
      .enter().append("circle")
      .attr("cx", d => x(d.Country) + x.bandwidth() / 2)
      .attr("cy", d => yRight(d.Infant_Mortality))
      .attr("r", 5)
      .attr("fill", "red")
      .on("mousemove", (event, d) => {
        const tooltipContent = `<strong>Country:</strong> ${d.Country}<br>
                                <strong>Infant Mortality:</strong> ${d.Infant_Mortality}`;
        showTooltip(tooltipContent, event);
      })
      .on("mouseout", hideTooltip);

    // X-Axis
    svg.append("g")
      .attr("transform", `translate(0,${height - margin.bottom})`)
      .call(d3.axisBottom(x))
      .selectAll("text")
      .attr("transform", "rotate(-45)")
      .style("text-anchor", "end");

    // Left Y-Axis for Life Expectancy
    svg.append("g")
      .attr("transform", `translate(${margin.left},0)`)
      .call(d3.axisLeft(yLeft).ticks(5))
      .append("text")
      .attr("fill", "black")
      .attr("x", -height / 2)
      .attr("dy", "-2.5em")
      .attr("text-anchor", "middle")
      .attr("transform", "rotate(-90)")
      .style("font-size", "12px")
      .text("Life Expectancy (Years)");

    // Right Y-Axis for Infant Mortality
    svg.append("g")
      .attr("transform", `translate(${width - margin.right},0)`)
      .call(d3.axisRight(yRight).ticks(5))
      .append("text")
      .attr("fill", "black")
      .attr("x", height / 2)
      .attr("dy", "2.5em")
      .attr("text-anchor", "middle")
      .attr("transform", "rotate(-90)")
      .style("font-size", "12px")
      .text("Infant Mortality (per 1000 live births)");

    // Title and Subtitle
    svg.append("text")
      .attr("x", width / 2)
      .attr("y", margin.top / 2)
      .attr("text-anchor", "middle")
      .style("font-size", "18px")
      .style("font-weight", "bold")
      .text("Dual-Axis Chart: Life Expectancy & Infant Mortality by Country");

    svg.append("text")
      .attr("x", width / 2)
      .attr("y", margin.top)
      .attr("text-anchor", "middle")
      .style("font-size", "12px")
      .style("fill", "gray")
      .text("Comparing Life Expectancy and Infant Mortality across countries");

    // Adding gridlines
    svg.append("g")
      .attr("class", "grid")
      .attr("transform", `translate(${margin.left},0)`)
      .call(d3.axisLeft(yLeft).ticks(5).tickSize(-width + margin.left + margin.right).tickFormat(""));
  });
}
```

Figure 14. Javascript for Dual-Axis Chart

Improvement Recommendations: For better clarity, think about using color-coding or marking particular countries that have the highest and lowest figures in life expectancy and infant mortality. Incorporating trend lines or categorizing countries by regions might also offer additional context and uncover patterns. Furthermore, increasing the visibility of the axes labels and ensuring color consistency will improve clarity and the overall visual effect.

Conclusion

Our visualization project aimed to shed light on significant global socio-economic metrics through an interactive and accessible website. By leveraging technologies such as HTML, CSS, D3.js, and JavaScript, we successfully transformed complex data into clear, compelling visual narratives that promote deeper understanding. This project emphasizes the critical connection between income inequality, life expectancy, and infant mortality rates, allowing users to draw meaningful insights.

As we reflect on our work, it’s evident that data visualization is a powerful tool to communicate complex information and prompt thoughtful analysis. We hope this project encourages further exploration and discussion around global disparities and inspires future work that expands on these findings to drive awareness and change.

Moving forward, integrating additional data points or real-time updates could provide even more dynamic insights, reinforcing the value of digital tools in interpreting and sharing critical information.

References

1. Indeed Editorial Team. (2023, July 31). Types of Graphs and Charts | Indeed.com. Www.indeed.com. https://www.indeed.com/career-advice/career-development/types-of-graphs-and-charts

2. W3Schools. (n.d.). HTML Tutorial. W3schools.com. https://www.w3schools.com/html/

3. Scatter | Chart.js. (2024). Chartjs.org. https://www.chartjs.org/docs/latest/samples/other-charts/scatter.html

4. Dixon & Moe. (2017). HTML Color Codes. HTML Color Codes. https://htmlcolorcodes.com/

5. Income inequality.(2024.OECD.https://www.oecd.org/en/data/indicators/income-inequality.html?oecdcontrol-8027380c62-var3=2021

6. Life expectancy at birth. (n.d.). OECD. https://www.oecd.org/en/data/indicators/life-expectancy-at-birth.html?oecdcontrol-f42fb73652-var3=2021

7. Infant mortality rates. (n.d.). OECD. https://www.oecd.org/en/data/indicators/infant-mortality-rates.html

Appendices