**Expt 8 : Designing Interactive Dashboards and Storytelling using D3.js on Environment/Forest Cover Dataset**

**Name: Anish Gade**

**UID: 2021700022**

Aim:

To design interactive dashboards and create visual storytelling using D3.js on a dataset related to Environment/Forest cover, covering basic and advanced charts.

Objectives:

1. **To understand how to use D3.js for data visualization.**
2. **To implement basic charts like Bar chart, Pie chart, Histogram, Timeline chart, Scatter plot, and Bubble plot.**
3. **To To implement advanced charts like Word chart, Box and whisker plot, Violin plot, Regression plot (linear and nonlinear), 3D chart, and Jitter.**
4. **draw observations and insights from each chart.**
5. **To create an interactive storytelling dashboard using the above visualizations.**

Expected Outcomes:

1. **Ability to create various types of visualizations using D3.js.**
2. **Interactive dashboards demonstrating different types of charts.**
3. **Insights from the Environment/Forest cover dataset through visual storytelling.**

Code:

<!DOCTYPE html>

<html>

<head>

    <title>Data Exploration Dashboard</title>

    <script src="https://d3js.org/d3.v6.min.js"></script>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 20px;

            background-color: #f5f5f5;

        }

        h1 {

            text-align: center;

            color: #333;

            padding: 20px 0;

            margin-bottom: 30px;

            background-color: #fff;

            box-shadow: 0 2px 4px rgba(0,0,0,0.1);

        }

        .dashboard-container {

            display: grid;

            grid-template-columns: repeat(2, 1fr);

            gap: 30px;

            max-width: 1400px;

            margin: 0 auto;

        }

        .chart-container {

            background: white;

            border-radius: 8px;

            padding: 20px;

            box-shadow: 0 2px 4px rgba(0,0,0,0.1);

        }

        .chart-container h2 {

            margin-top: 0;

            margin-bottom: 20px;

            color: #444;

            font-size: 1.2em;

            text-align: center;

        }

        /\* Make charts responsive \*/

        .chart {

            width: 100%;

            height: 400px;

            overflow: hidden;

        }

        /\* Axis styling \*/

        .axis-label {

            font-size: 12px;

            fill: #666;

        }

        .axis path,

        .axis line {

            stroke: #ccc;

        }

        /\* Responsive design \*/

        @media (max-width: 1200px) {

            .dashboard-container {

                grid-template-columns: 1fr;

            }

        }

        /\* Tooltip styling \*/

        .tooltip {

            position: absolute;

            padding: 8px;

            background: rgba(0, 0, 0, 0.8);

            color: white;

            border-radius: 4px;

            font-size: 12px;

            pointer-events: none;

        }

    </style>

</head>

<body>

    <h1>Plastic Waste Analysis Dashboard</h1>

    <div class="dashboard-container">

        <div class="chart-container">

            <h2>Top 10 Countries by Plastic Waste</h2>

            <div id="bar-chart" class="chart"></div>

        </div>

        <div class="chart-container">

            <h2>Distribution of Waste Sources</h2>

            <div id="pie-chart" class="chart"></div>

        </div>

        <div class="chart-container">

            <h2>Recycling Rate vs. Per Capita Waste</h2>

            <div id="scatter-plot" class="chart"></div>

        </div>

        <div class="chart-container">

            <h2>Waste Impact Analysis</h2>

            <div id="bubble-chart" class="chart"></div>

        </div>

    </div>

    <script>

        // Create tooltip div

        const tooltip = d3.select("body")

            .append("div")

            .attr("class", "tooltip")

            .style("opacity", 0);

        d3.csv('data.csv').then(function(data) {

            // Sort data by Total\_Plastic\_Waste\_MT and get top 10

            const top10Data = data

                .sort((a, b) => b.Total\_Plastic\_Waste\_MT - a.Total\_Plastic\_Waste\_MT)

                .slice(0, 10);

            createBarChart(top10Data);

            createPieChart(data);

            createScatterPlot(data);

            createBubbleChart(data);

        });

        function createBarChart(data) {

            const margin = {top: 20, right: 20, bottom: 60, left: 60};

            const width = document.getElementById('bar-chart').clientWidth - margin.left - margin.right;

            const height = 400 - margin.top - margin.bottom;

            const svg = d3.select("#bar-chart")

                .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})`);

            const x = d3.scaleBand()

                .range([0, width])

                .padding(0.1);

            const y = d3.scaleLinear()

                .range([height, 0]);

            x.domain(data.map(d => d.Country));

            y.domain([0, d3.max(data, d => +d.Total\_Plastic\_Waste\_MT)]);

            // Add bars

            svg.selectAll(".bar")

                .data(data)

                .enter()

                .append("rect")

                .attr("class", "bar")

                .attr("x", d => x(d.Country))

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

                .attr("y", d => y(d.Total\_Plastic\_Waste\_MT))

                .attr("height", d => height - y(d.Total\_Plastic\_Waste\_MT))

                .attr("fill", "steelblue")

                .on("mouseover", function(event, d) {

                    tooltip.transition()

                        .duration(200)

                        .style("opacity", .9);

                    tooltip.html(`${d.Country}<br/>${d.Total\_Plastic\_Waste\_MT} MT`)

                        .style("left", (event.pageX) + "px")

                        .style("top", (event.pageY - 28) + "px");

                })

                .on("mouseout", function(d) {

                    tooltip.transition()

                        .duration(500)

                        .style("opacity", 0);

                });

            // Add X axis

            svg.append("g")

                .attr("transform", `translate(0,${height})`)

                .call(d3.axisBottom(x))

                .selectAll("text")

                .attr("transform", "rotate(-45)")

                .style("text-anchor", "end");

            // Add Y axis

            svg.append("g")

                .call(d3.axisLeft(y));

            // Add labels

            svg.append("text")

                .attr("class", "axis-label")

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

                .attr("x", width/2)

                .attr("y", height + margin.bottom - 5)

                .text("Country");

            svg.append("text")

                .attr("class", "axis-label")

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

                .attr("transform", "rotate(-90)")

                .attr("y", -margin.left + 20)

                .attr("x", -height/2)

                .text("Total Plastic Waste (MT)");

        }

        function createPieChart(data) {

            const width = document.getElementById('pie-chart').clientWidth;

            const height = 400;

            const radius = Math.min(width, height) / 2 - 40;

            const svg = d3.select("#pie-chart")

                .append("svg")

                .attr("width", width)

                .attr("height", height)

                .append("g")

                .attr("transform", `translate(${width/2},${height/2})`);

            // Aggregate data by Main\_Sources

            const sourceData = Array.from(d3.group(data, d => d.Main\_Sources),

                ([key, value]) => ({

                    source: key,

                    count: value.length

                }));

            const color = d3.scaleOrdinal()

                .domain(sourceData.map(d => d.source))

                .range(d3.schemeCategory10);

            const pie = d3.pie()

                .value(d => d.count);

            const arc = d3.arc()

                .innerRadius(0)

                .outerRadius(radius);

            // Add slices

            const arcs = svg.selectAll("arc")

                .data(pie(sourceData))

                .enter()

                .append("g");

            arcs.append("path")

                .attr("d", arc)

                .attr("fill", d => color(d.data.source))

                .attr("stroke", "white")

                .style("stroke-width", "2px")

                .on("mouseover", function(event, d) {

                    tooltip.transition()

                        .duration(200)

                        .style("opacity", .9);

                    tooltip.html(`${d.data.source}<br/>${d.data.count} countries`)

                        .style("left", (event.pageX) + "px")

                        .style("top", (event.pageY - 28) + "px");

                })

                .on("mouseout", function(d) {

                    tooltip.transition()

                        .duration(500)

                        .style("opacity", 0);

                });

            // Add labels

            const labelArc = d3.arc()

                .innerRadius(radius \* 0.6)

                .outerRadius(radius \* 0.6);

            arcs.append("text")

                .attr("transform", d => `translate(${labelArc.centroid(d)})`)

                .attr("dy", "0.35em")

                .text(d => d.data.source)

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

                .style("font-size", "12px")

                .style("fill", "#fff");

        }

        function createScatterPlot(data) {

            const margin = {top: 20, right: 20, bottom: 60, left: 60};

            const width = document.getElementById('scatter-plot').clientWidth - margin.left - margin.right;

            const height = 400 - margin.top - margin.bottom;

            const svg = d3.select("#scatter-plot")

                .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})`);

            const x = d3.scaleLinear()

                .domain([0, d3.max(data, d => +d.Recycling\_Rate)])

                .range([0, width]);

            const y = d3.scaleLinear()

                .domain([0, d3.max(data, d => +d.Per\_Capita\_Waste\_KG)])

                .range([height, 0]);

            // Add dots

            svg.selectAll("dot")

                .data(data)

                .enter()

                .append("circle")

                .attr("cx", d => x(d.Recycling\_Rate))

                .attr("cy", d => y(d.Per\_Capita\_Waste\_KG))

                .attr("r", 5)

                .style("fill", d => {

                    switch(d.Coastal\_Waste\_Risk) {

                        case "Very\_High": return "#ff0000";

                        case "High": return "#ff9900";

                        case "Medium": return "#ffff00";

                        default: return "#00ff00";

                    }

                })

                .style("opacity", 0.7)

                .on("mouseover", function(event, d) {

                    tooltip.transition()

                        .duration(200)

                        .style("opacity", .9);

                    tooltip.html(`${d.Country}<br/>Recycling: ${d.Recycling\_Rate}%<br/>Per Capita: ${d.Per\_Capita\_Waste\_KG} kg`)

                        .style("left", (event.pageX) + "px")

                        .style("top", (event.pageY - 28) + "px");

                })

                .on("mouseout", function(d) {

                    tooltip.transition()

                        .duration(500)

                        .style("opacity", 0);

                });

            // Add X axis

            svg.append("g")

                .attr("transform", `translate(0,${height})`)

                .call(d3.axisBottom(x));

            // Add Y axis

            svg.append("g")

                .call(d3.axisLeft(y));

            // Add labels

            svg.append("text")

                .attr("class", "axis-label")

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

                .attr("x", width/2)

                .attr("y", height + margin.bottom - 5)

                .text("Recycling Rate (%)");

            svg.append("text")

                .attr("class", "axis-label")

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

                .attr("transform", "rotate(-90)")

                .attr("y", -margin.left + 20)

                .attr("x", -height/2)

                .text("Per Capita Waste (kg)");

        }

        function createBubbleChart(data) {

            const margin = {top: 20, right: 20, bottom: 60, left: 60};

            const width = document.getElementById('bubble-chart').clientWidth - margin.left - margin.right;

            const height = 400 - margin.top - margin.bottom;

            const svg = d3.select("#bubble-chart")

                .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})`);

            const x = d3.scaleLinear()

                .domain([0, d3.max(data, d => +d.Total\_Plastic\_Waste\_MT)])

                .range([0, width]);

            const y = d3.scaleLinear()

                .domain([0, d3.max(data, d => +d.Per\_Capita\_Waste\_KG)])

                .range([height, 0]);

            const radius = d3.scaleSqrt()

                .domain([0, d3.max(data, d => +d.Total\_Plastic\_Waste\_MT)])

                .range([4, 40]);

            // Add bubbles

            svg.selectAll("circle")

                .data(data)

                .enter()

                .append("circle")

                .attr("cx", d => x(d.Total\_Plastic\_Waste\_MT))

                .attr("cy", d => y(d.Per\_Capita\_Waste\_KG))

                .attr("r", d => radius(d.Total\_Plastic\_Waste\_MT))

                .style("fill", d => {

                    switch(d.Coastal\_Waste\_Risk) {

                        case "Very\_High": return "#ff0000";

                        case "High": return "#ff9900";

                        case "Medium": return "#ffff00";

                        default: return "#00ff00";

                    }

                })

                .style("opacity", 0.7)

                .on("mouseover", function(event, d) {

                    tooltip.transition()

                        .duration(200)

                        .style("opacity", .9);

                    tooltip.html(`${d.Country}<br/>Total Waste: ${d.Total\_Plastic\_Waste\_MT} MT<br/>Per Capita: ${d.Per\_Capita\_Waste\_KG} kg<br/>Risk: ${d.Coastal\_Waste\_Risk}`)

                        .style("left", (event.pageX) + "px")

                        .style("top", (event.pageY - 28) + "px");

                })

                .on("mouseout", function(d) {

                    tooltip.transition()

                        .duration(500)

                        .style("opacity", 0);

                });

            // Add X axis

            svg.append("g")

                .attr("transform", `translate(0,${height})`)

                .call(d3.axisBottom(x));

            // Add Y axis

            svg.append("g")

                .call(d3.axisLeft(y));

            // Add labels

            svg.append("text")

                .attr("class", "axis-label")

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

                .attr("x", width/2)

                .attr("y", height + margin.bottom - 5)

                .text("Total Plastic Waste (MT)");

            svg.append("text")

                .attr("class", "axis-label")

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

                .attr("transform", "rotate(-90)")

                .attr("y", -margin.left + 20)

                .attr("x", -height/2)

                .text("Per Capita Waste (kg)");

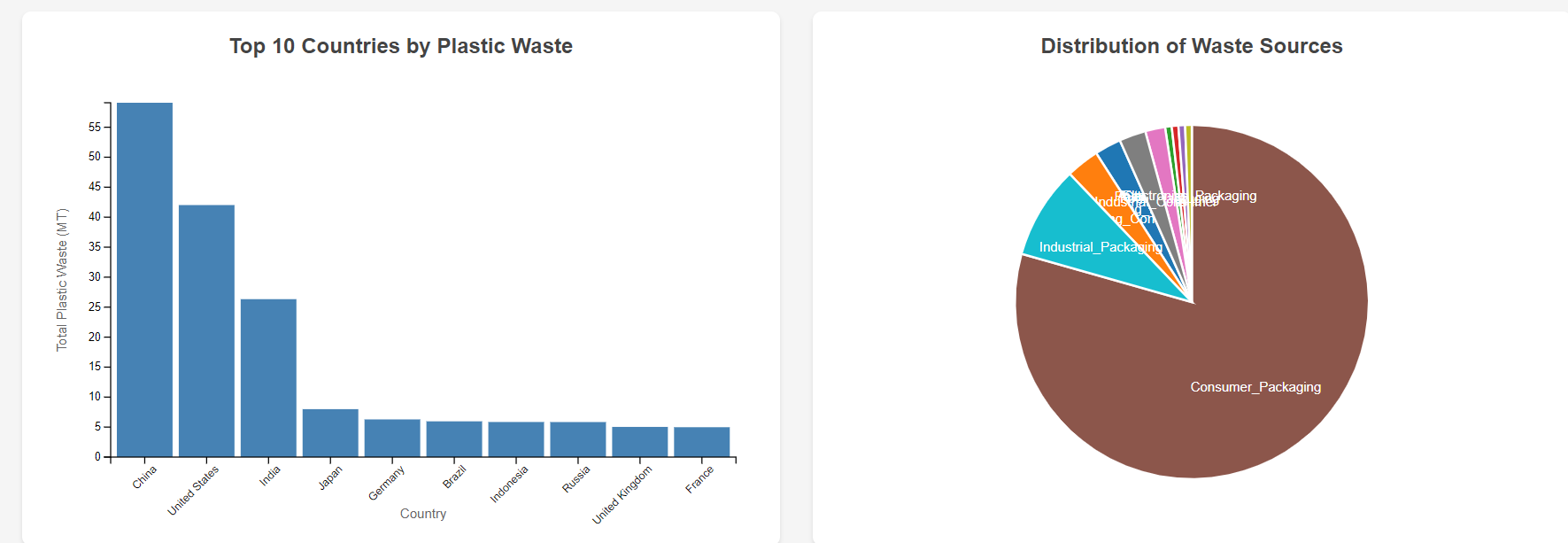
        }

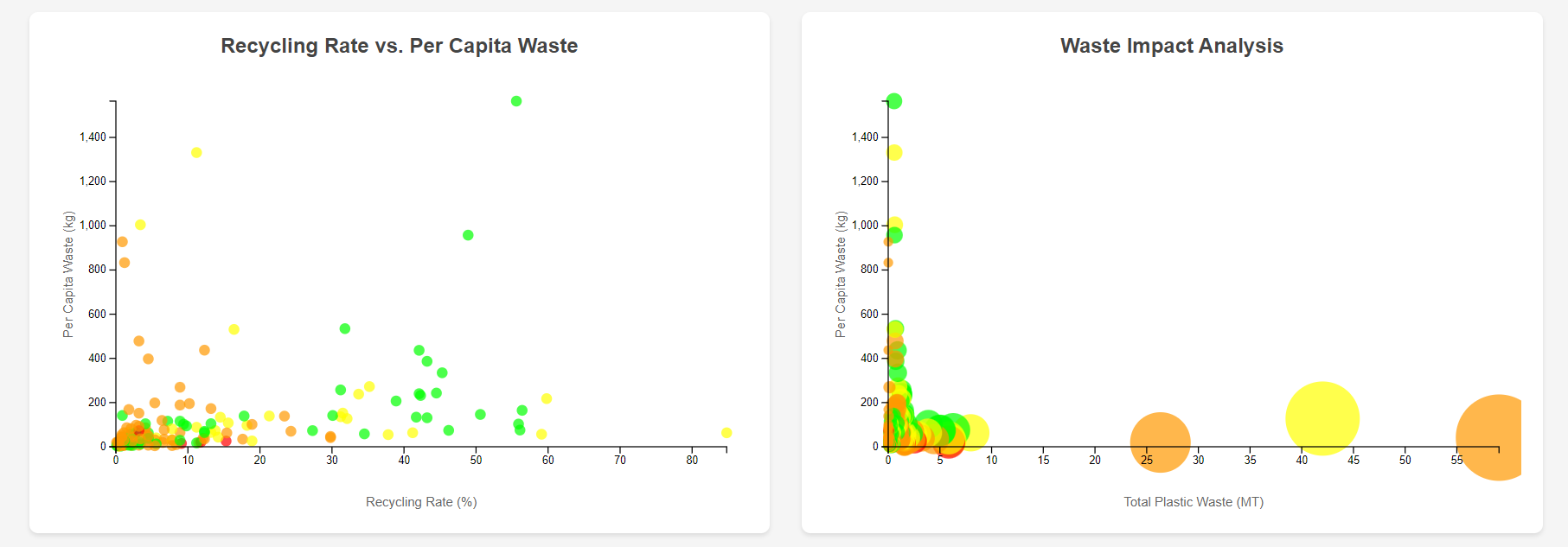
    </script>

</body>

</html>

Output:





Summary:

In this lab, we created an interactive dashboard using D3.js for data visualization on an Environment/Forest cover dataset. We explored both basic and advanced chart types and built a storytelling dashboard to provide insights into forest cover trends and distributions.

Conclusion:

D3.js is a powerful library for creating dynamic, interactive data visualizations. By implementing various chart types, we could extract meaningful insights about forest cover, trends, and patterns.