Sardar Patel Institute of Technology,Mumbai
Department of Electronics and Telecommunication Engineering
B.E. Sem-VII- PE-IV (2024-2025)

Name :- Manthan Ayalwar
Batch :- ADV I
Uid :- 2021700003
Branch :- Cse-Ds

Experiment no 7

Aim :

Experiment Design for Creating Visualizations using D3.js on a Finance Dataset

Objectives:

- **To explore and visualize a dataset related to Finance/Banking/Insurance/Credit using D3.js.**
- **To create basic visualizations (Bar chart, Pie chart, Histogram, Timeline chart, Scatter plot, Bubble plot) to understand data distribution and trends.**
- **To create advanced visualizations (Word chart, Box and Whisker plot, Violin plot, Regression plot, 3D chart, Jitter) for deeper insights and complex relationships.**

Database:
https://www.kaggle.com/datasets/mhdzahier/travel-insurance

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>D3 Visualization</title>

  <!-- Link CSS file -->
  <link rel="stylesheet" href="styles.css">
```

```
  <!-- D3.js Library -->
  <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
  <h2>Net Sales per Agency</h2>

  <!-- SVG Canvas -->
  <svg width="800" height="500"></svg>

  <!-- Link external JavaScript file -->
  <script src="pie_chart.js"></script>
</body>
</html>
```

```
/* SVG and chart styles */
svg {
    font-family: Arial, sans-serif;
  }

  /* Styling for the bars */
  .bar {
    fill: steelblue;
  }

  /* On hover, change bar color */
  .bar:hover {
    fill: orange;
  }

  /* Styling for axis labels */
  .axis-label {
    font-size: 12px;
  }
```

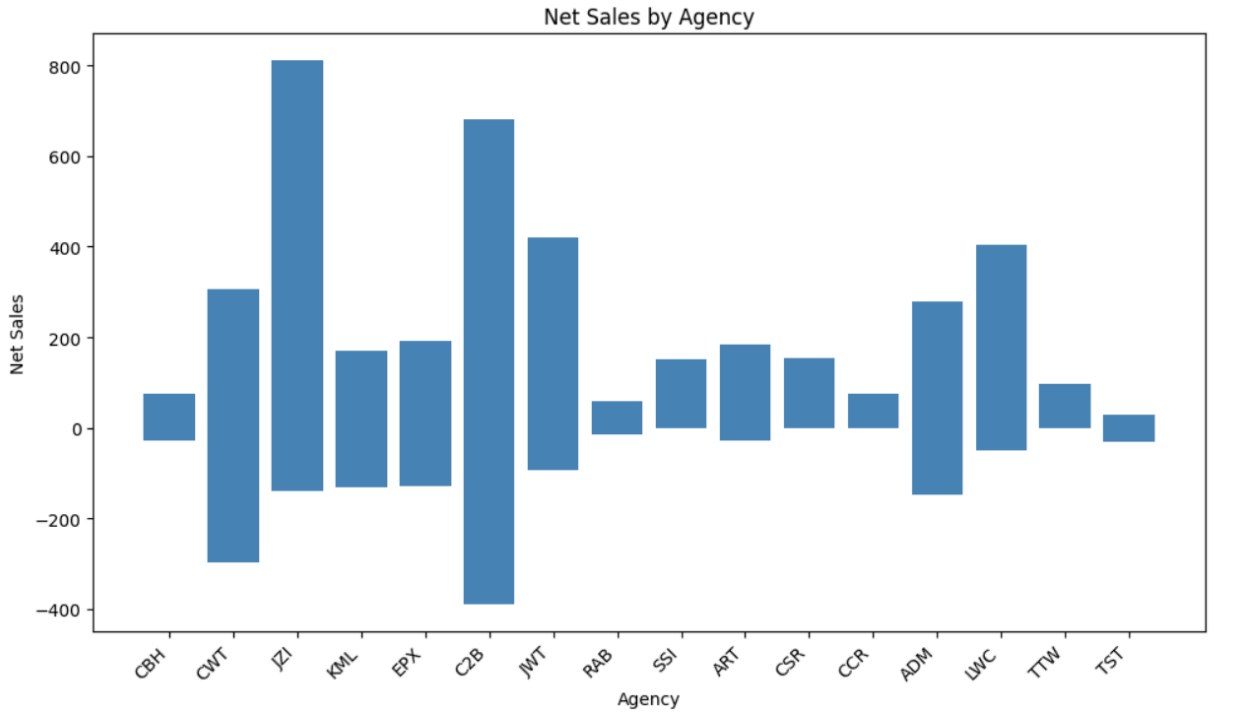
```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pearson Correlation Coefficient</title>
    <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
    <h1>Pearson Correlation Coefficient</h1>
    <div id="result"></div>
    <script>
        // Load the dataset
        d3.csv("travel_insurance.csv").then(data => {
            // Convert data types
            data.forEach(d => {
                d.Age = +d.Age; // Assuming Age is a numerical variable
                d["Net Sales"] = +d["Net Sales"]; // Assuming Net Sales is
a numerical variable
            });

            // Calculate the Pearson correlation coefficient
            const { r, pValue } = pearsonCorrelation(data.map(d => d.Age),
data.map(d => d["Net Sales"]));

            // Display the results
            d3.select("#result").html(`Pearson Correlation Coefficient:
${r.toFixed(4)}<br>P-value: ${pValue.toFixed(4)}`);
        });

        // Function to calculate Pearson correlation coefficient and
p-value
        function pearsonCorrelation(x, y) {
            const n = x.length;
            if (n !== y.length) throw new Error("Input arrays must have
the same length.");

            const sumX = d3.sum(x);
            const sumY = d3.sum(y);
            const sumX2 = d3.sum(x.map(i => i * i));
            const sumY2 = d3.sum(y.map(i => i * i));
            const sumXY = d3.sum(x.map((xi, i) => xi * y[i]));
```

```
            const numerator = n * sumXY - sumX * sumY;
            const denominator = Math.sqrt((n * sumX2 - sumX * sumX) * (n *
sumY2 - sumY * sumY));

            const r = numerator / denominator; // Pearson correlation
coefficient

            // Calculate p-value using t-distribution
            const t = (r * Math.sqrt(n - 2)) / Math.sqrt(1 - r * r);
            const pValue = 2 * (1 - jStat.studentt.cdf(Math.abs(t), n -
2)); // Using jStat library for p-value calculation

            return { r, pValue };
        }
    </script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jstat/1.9.1/jstat.min.js"></sc
ript> <!-- Include jStat for statistical functions -->
</body>
</html>
```

Visualization:

1] Bar Chart

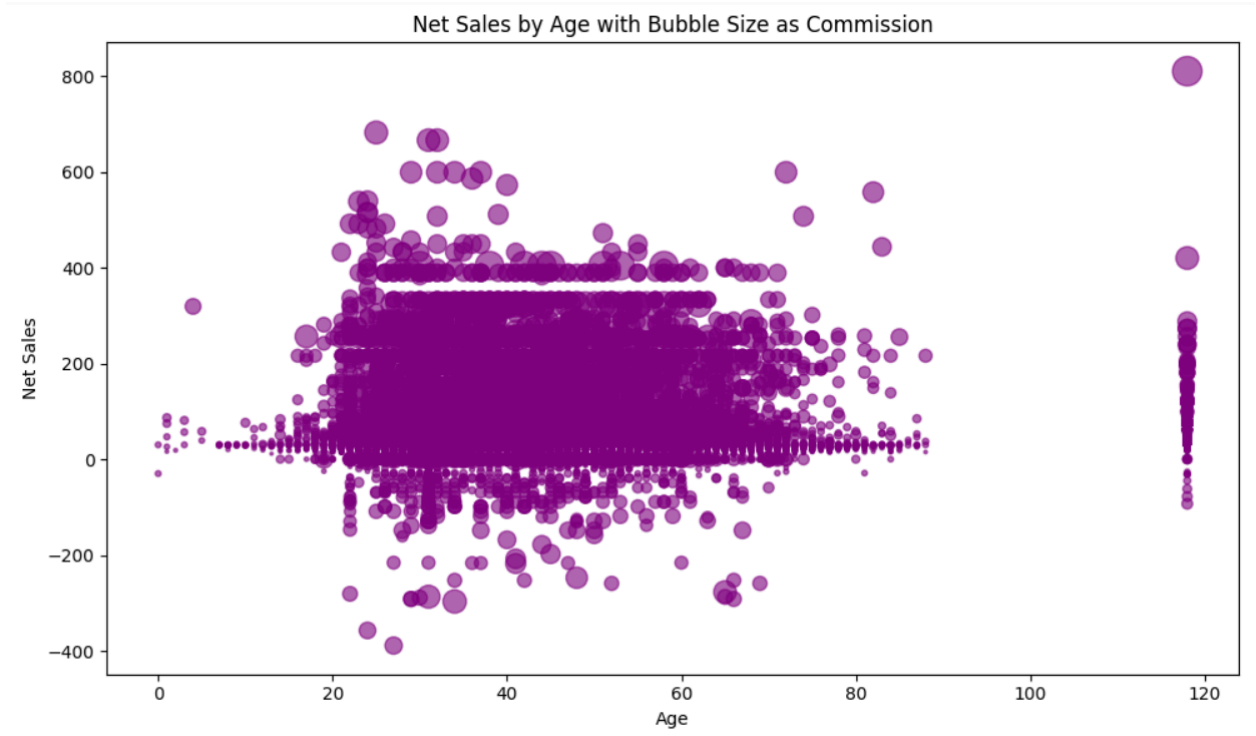



```
d3.csv("travel_insurance.csv").then(data => {

    const svg = d3.select("svg");

    svg.selectAll("*").remove(); // Clear previous chart



    const width = +svg.attr("width");

    const height = +svg.attr("height");

    const margin = { top: 40, right: 20, bottom: 70, left: 60 };



    const xScale = d3.scaleBand()

      .domain(data.map(d => d.Agency))

      .range([margin.left, width - margin.right])

      .padding(0.2);
```

```
const yScale = d3.scaleLinear()
  .domain([0, d3.max(data, d => +d["Net Sales"])])
  .range([height - margin.bottom, margin.top]);

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

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

svg.selectAll(".bar")
  .data(data)
  .enter()
  .append("rect")
  .attr("class", "bar")
  .attr("x", d => xScale(d.Agency))
  .attr("y", d => yScale(+d["Net Sales"]))
```

```
      .attr("width", xScale.bandwidth())

      .attr("height", d => height - margin.bottom - yScale(+d["Net
Sales"]))

      .attr("fill", "steelblue");

  });
```

2] Bubble Plot

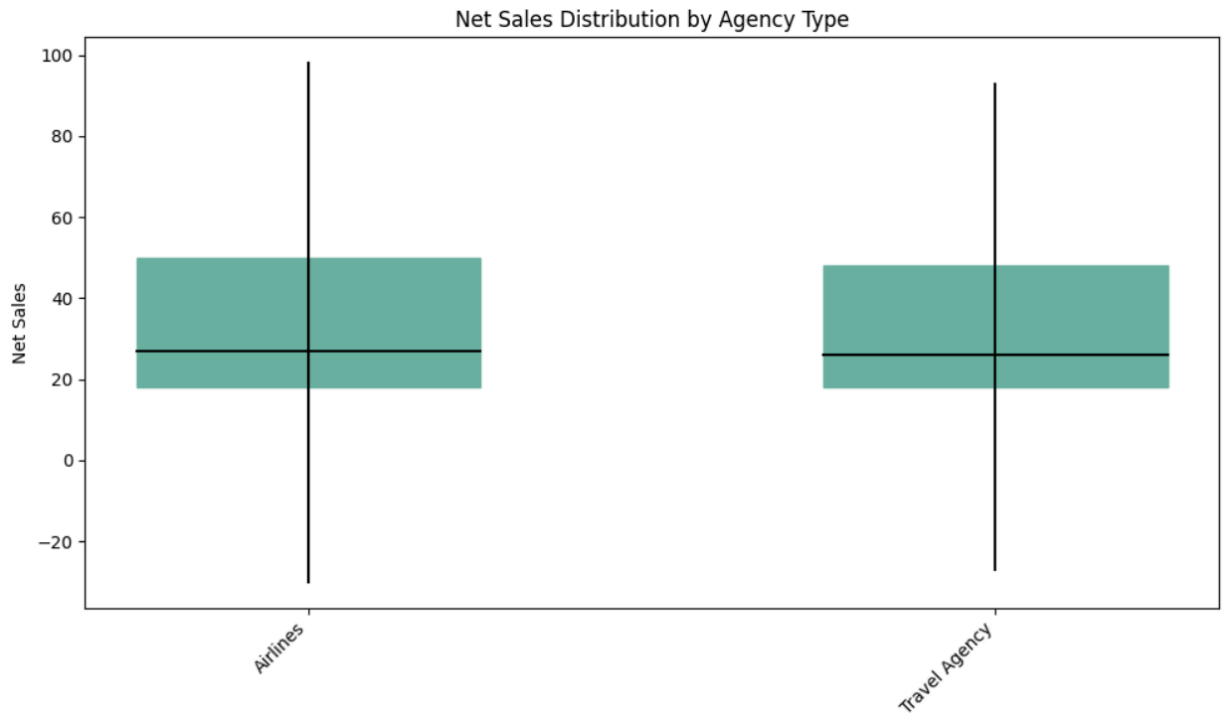



```
d3.csv("travel_insurance.csv").then(data => {

    const svg = d3.select("svg");

    svg.selectAll("*").remove();



    const width = +svg.attr("width");
```

```
    const height = +svg.attr("height");

    const margin = { top: 20, right: 20, bottom: 40, left: 40 };



    const xScale = d3.scaleLinear()

      .domain([0, d3.max(data, d => +d.Age)])

      .range([margin.left, width - margin.right]);



    const yScale = d3.scaleLinear()

      .domain([d3.min(data, d => +d["Net Sales"]), d3.max(data, d =>
+d["Net Sales"])])

      .range([height - margin.bottom, margin.top]);



    const radiusScale = d3.scaleSqrt()

      .domain([0, d3.max(data, d => +d["Commision (in value)"])])

      .range([2, 20]);



    svg.append("g")

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

      .call(d3.axisBottom(xScale));



    svg.append("g")

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

      .call(d3.axisLeft(yScale));
```

```
    svg.selectAll("circle")

      .data(data)

      .enter()

      .append("circle")

      .attr("cx", d => xScale(+d.Age))

      .attr("cy", d => yScale(+d["Net Sales"]))

      .attr("r", d => radiusScale(+d["Commision (in value)"]))

      .attr("fill", "purple")

      .attr("opacity", 0.6)

      .append("title")

      .text(d => `Age: ${d.Age}, Net Sales: ${d["Net Sales"]}, Commission:
${d["Commision (in value)"]}`);

  });
```

3] Box and Whisker



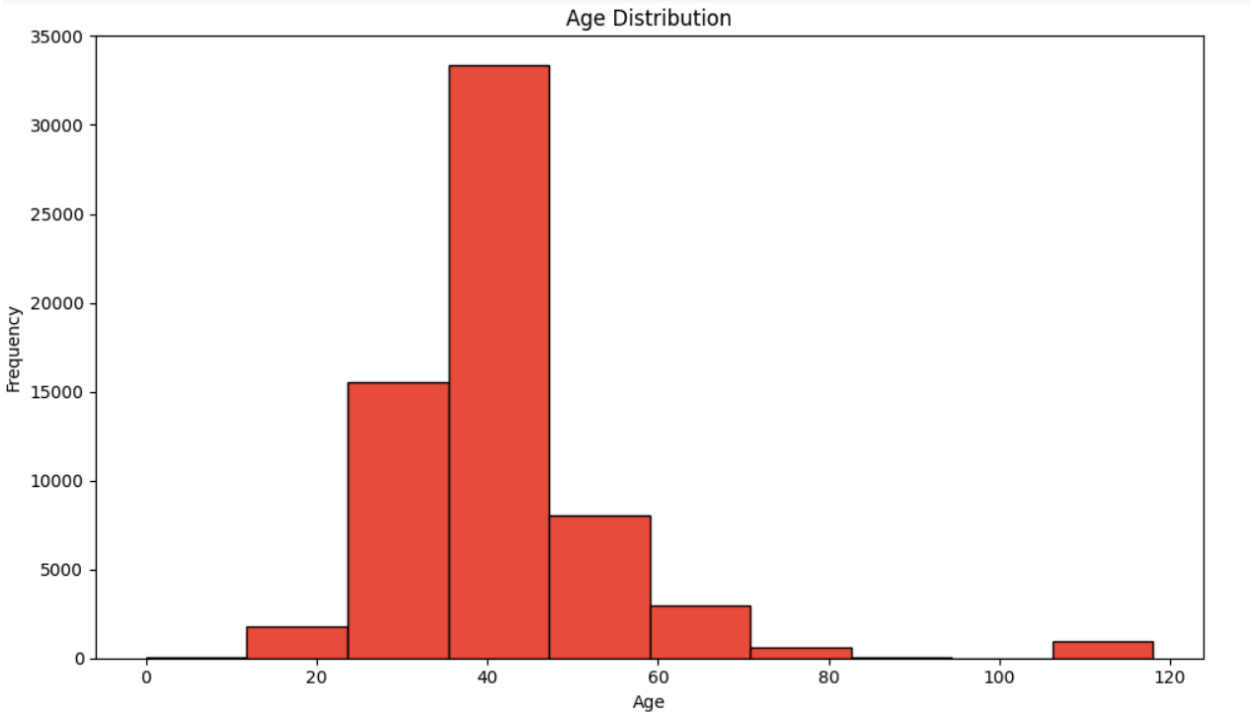

```
d3.csv("travel_insurance.csv").then(data => {

    const svg = d3.select("svg");

    svg.selectAll("*").remove();



    const width = +svg.attr("width");

    const height = +svg.attr("height");

    const margin = { top: 20, right: 30, bottom: 40, left: 50 };



    const agencyTypes = Array.from(new Set(data.map(d => d["Agency
Type"])));



    const xScale = d3.scaleBand()

      .domain(agencyTypes)
```

```
      .range([margin.left, width - margin.right])
      .padding(0.5);

    const yScale = d3.scaleLinear()
      .domain(d3.extent(data, d => +d["Net Sales"]))
      .range([height - margin.bottom, margin.top]);

    const groupedData = d3.groups(data, d => d["Agency Type"]).map(([key, values]) => {
      const sales = values.map(d => +d["Net Sales"]);
      sales.sort(d3.ascending);
      const q1 = d3.quantile(sales, 0.25);
      const median = d3.quantile(sales, 0.5);
      const q3 = d3.quantile(sales, 0.75);
      const iqr = q3 - q1;
      const min = Math.max(d3.min(sales), q1 - 1.5 * iqr);
      const max = Math.min(d3.max(sales), q3 + 1.5 * iqr);
      return { key, q1, median, q3, min, max };
    });

    svg.append("g")
      .attr("transform", `translate(0,${height - margin.bottom})`)
      .call(d3.axisBottom(xScale));
```

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


  const boxWidth = xScale.bandwidth() / 2;


  svg.selectAll("g.box")
    .data(groupedData)
    .enter().append("g")
    .attr("transform", d => `translate(${xScale(d.key)}, 0)`)
    .each(function (d) {
      const g = d3.select(this);


      g.append("line")
        .attr("x1", 0)
        .attr("x2", 0)
        .attr("y1", yScale(d.min))
        .attr("y2", yScale(d.max))
        .attr("stroke", "black");


      g.append("rect")
        .attr("x", -boxWidth / 2)
        .attr("width", boxWidth)
```

```
          .attr("y", yScale(d.q3))
          .attr("height", yScale(d.q1) - yScale(d.q3))
          .attr("fill", "#69b3a2");


        g.append("line")
          .attr("x1", -boxWidth / 2)
          .attr("x2", boxWidth / 2)
          .attr("y1", yScale(d.median))
          .attr("y2", yScale(d.median))
          .attr("stroke", "black");
      });
});
```

4] Histogram

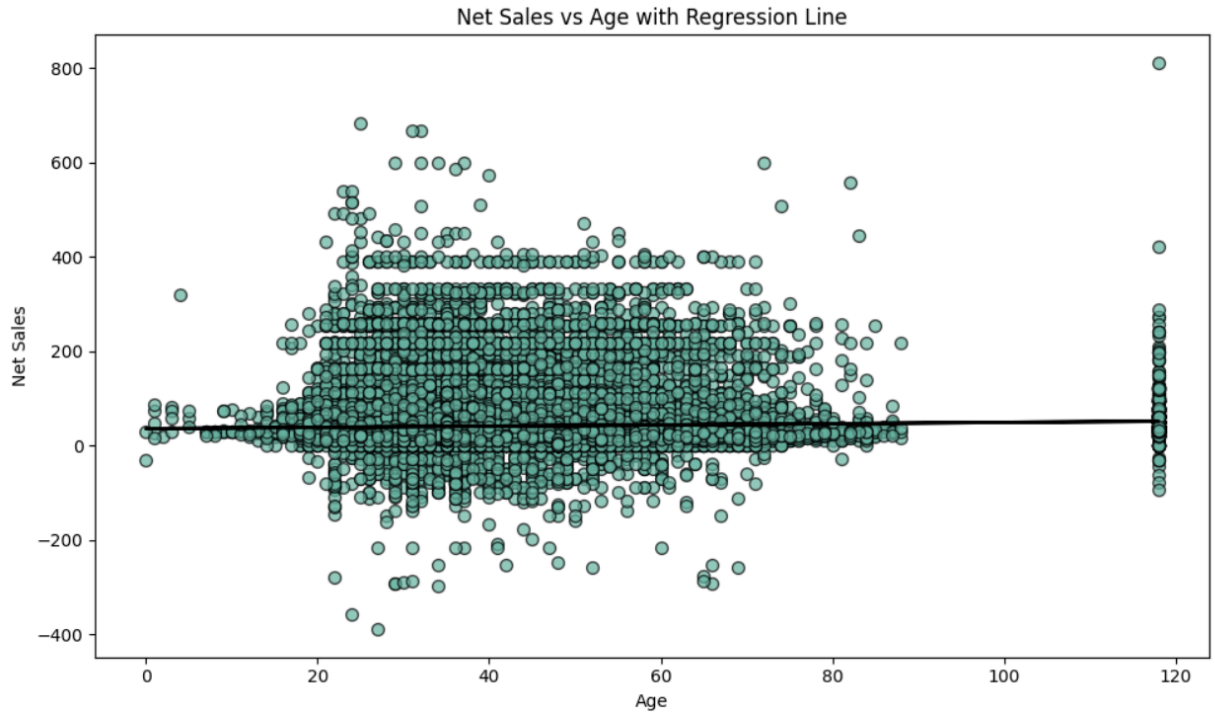



```
d3.csv("travel_insurance.csv").then(data => {

    const svg = d3.select("svg");

    svg.selectAll("*").remove();



    const width = +svg.attr("width");

    const height = +svg.attr("height");

    const margin = { top: 30, right: 20, bottom: 40, left: 40 };



    const xScale = d3.scaleLinear()

      .domain([0, d3.max(data, d => +d.Age)])

      .range([margin.left, width - margin.right]);



    const histogram = d3.histogram()
```

```
    .value(d => +d.Age)
    .domain(xScale.domain())
    .thresholds(xScale.ticks(10));

const bins = histogram(data);

const yScale = d3.scaleLinear()
    .domain([0, d3.max(bins, d => d.length)])
    .range([height - margin.bottom, margin.top]);

svg.append("g")
    .attr("transform", `translate(0,${height - margin.bottom})`)
    .call(d3.axisBottom(xScale));

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

svg.selectAll("rect")
    .data(bins)
    .enter()
    .append("rect")
    .attr("x", d => xScale(d.x0))
```

```
      .attr("y", d => yScale(d.length))

      .attr("width", d => xScale(d.x1) - xScale(d.x0) - 1)

      .attr("height", d => height - margin.bottom - yScale(d.length))

      .attr("fill", "#e74c3c");

  });
```

5] Regression Plot

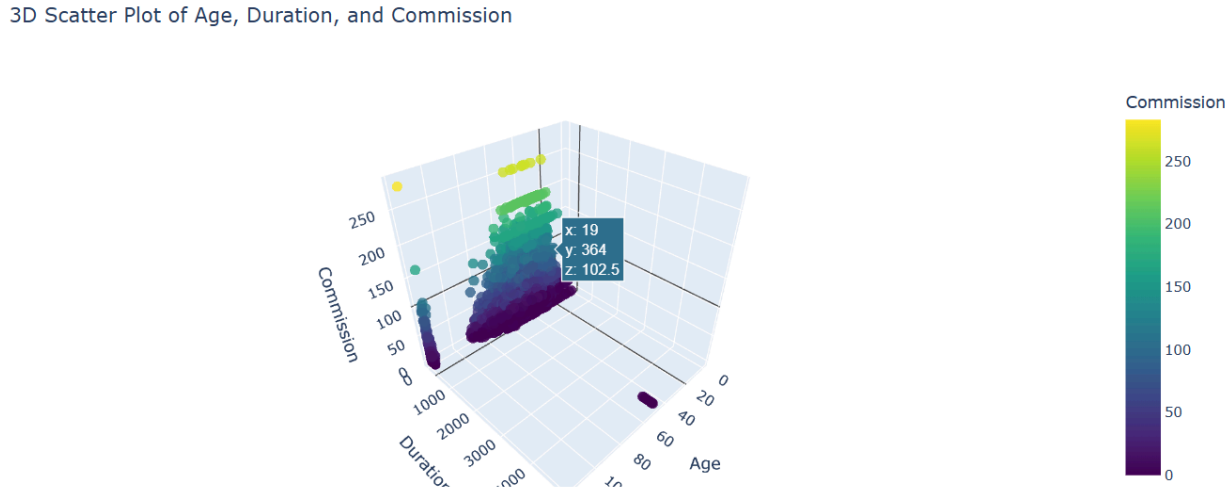



```
d3.csv("travel_insurance.csv").then(data => {

    const svg = d3.select("svg");

    svg.selectAll("*").remove();



    const width = +svg.attr("width");
```

```
    const height = +svg.attr("height");
    const margin = { top: 20, right: 30, bottom: 50, left: 50 };

    const xScale = d3.scaleLinear()
      .domain(d3.extent(data, d => +d.Age))
      .range([margin.left, width - margin.right]);

    const yScale = d3.scaleLinear()
      .domain(d3.extent(data, d => +d["Net Sales"]))
      .range([height - margin.bottom, margin.top]);

    svg.append("g")
      .attr("transform", `translate(0,${height - margin.bottom})`)
      .call(d3.axisBottom(xScale).ticks(10));

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

    const points = data.map(d => [xScale(+d.Age), yScale(+d["Net
Sales"])]);

    // Add scatter plot points
    svg.selectAll("circle")
```

```
    .data(data)
    .enter().append("circle")
    .attr("cx", d => xScale(+d.Age))
    .attr("cy", d => yScale(+d["Net Sales"]))
    .attr("r", 5)
    .attr("fill", "#69b3a2");

  // Perform linear regression
  const regression = d3.regressionLinear()
    .x(d => +d.Age)
    .y(d => +d["Net Sales"])(data);

  // Draw regression line
  svg.append("path")
    .datum(regression)
    .attr("fill", "none")
    .attr("stroke", "black")
    .attr("stroke-width", 2)
    .attr("d", d3.line()
      .x(d => xScale(d[0]))
      .y(d => yScale(d[1])));
});
```

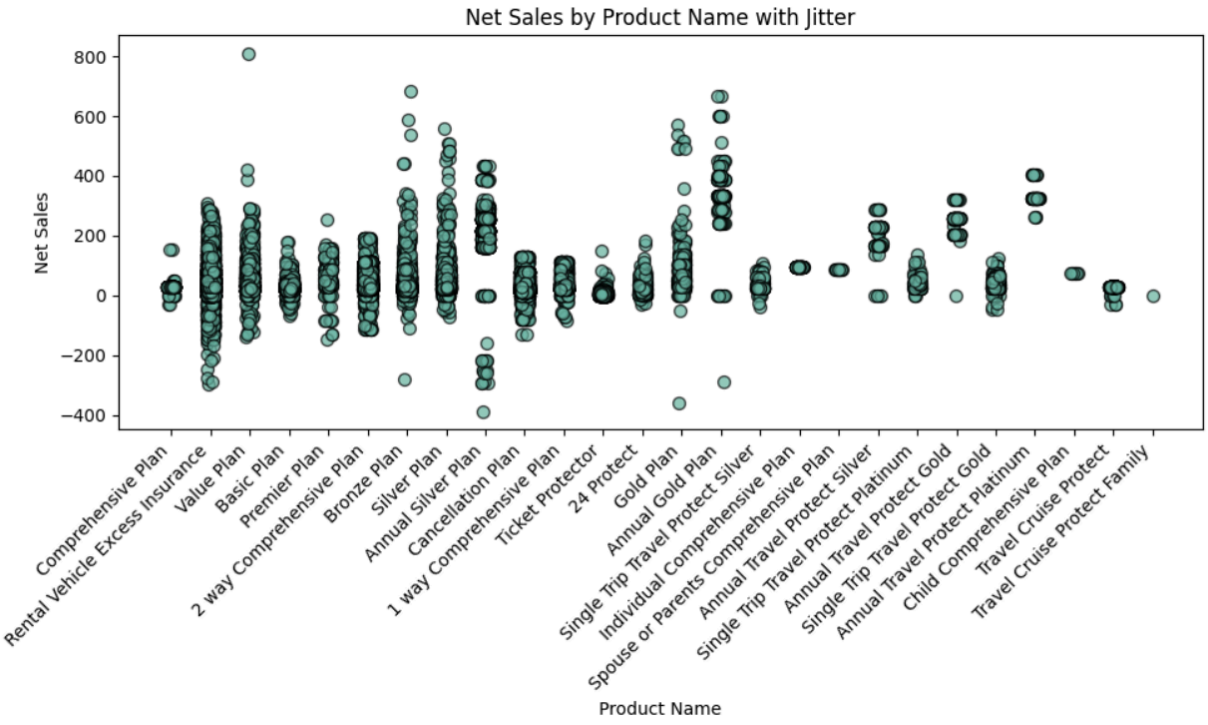
6] 3D Chart




```
Plotly.d3.csv("travel_insurance.csv", function(data) {

    const age = data.map(d => +d.Age);

    const duration = data.map(d => +d.Duration);

    const commission = data.map(d => +d["Commision (in value)"]);



    const trace = {

      x: age,

      y: duration,

      z: commission,

      mode: 'markers',

      marker: {

        size: 5,

        color: commission,

        colorscale: 'Viridis',
```

```
        showscale: true
      },
      type: 'scatter3d'
    };

    const layout = {
      title: '3D Scatter Plot of Age, Duration, and Commission',
      autosize: true
    };

    Plotly.newPlot('chart', [trace], layout);
});
```

7] Jitter Plot




```
d3.csv("travel_insurance.csv").then(data => {

    const svg = d3.select("svg");

    svg.selectAll("*").remove();



    const width = +svg.attr("width");

    const height = +svg.attr("height");

    const margin = { top: 20, right: 30, bottom: 100, left: 50 };



    const productNames = Array.from(new Set(data.map(d => d["Product
Name"])));



    const xScale = d3.scaleBand()

      .domain(productNames)
```

```
      .range([margin.left, width - margin.right])
      .padding(0.5);

    const yScale = d3.scaleLinear()
      .domain(d3.extent(data, d => +d["Net Sales"]))
      .range([height - margin.bottom, margin.top]);

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

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

    svg.selectAll("circle")
      .data(data)
      .enter().append("circle")
      .attr("cx", d => xScale(d["Product Name"]) + xScale.bandwidth() / 2
+ Math.random() * 10 - 5)
      .attr("cy", d => yScale(+d["Net Sales"]))
```

```
      .attr("r", 5)

      .attr("fill", "#69b3a2");

  });
```

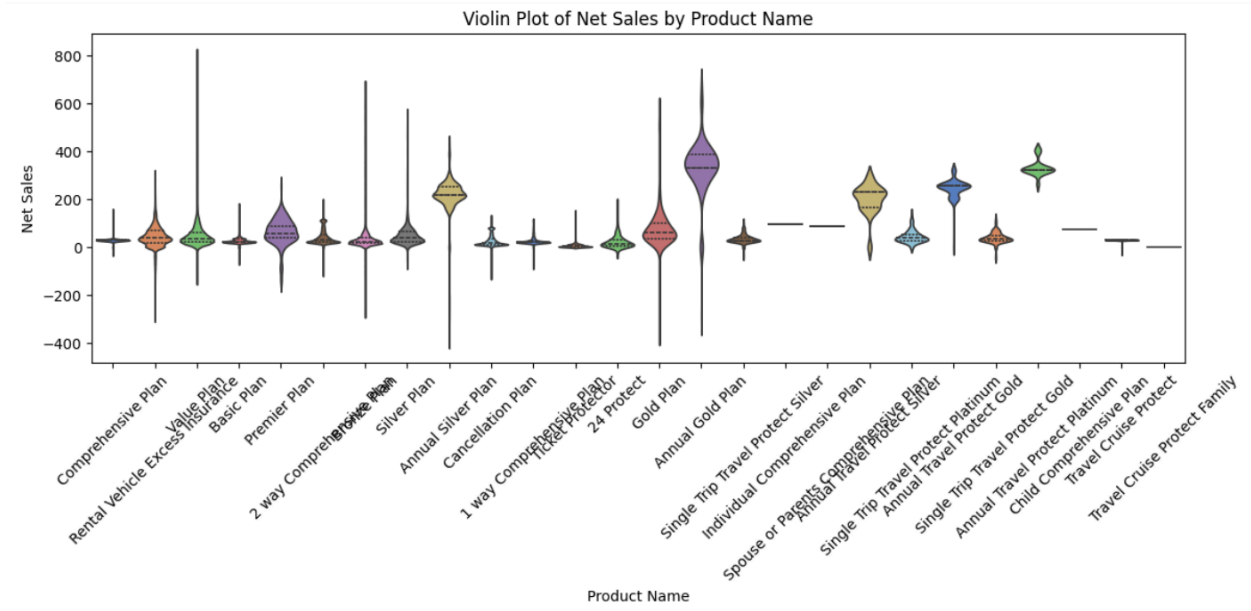
8] Word Chart


```
d3.csv("travel_insurance.csv").then(data => {

    const svg = d3.select("svg");

    svg.selectAll("*").remove();



    const width = +svg.attr("width");

    const height = +svg.attr("height");



    const destinationCounts = d3.rollup(
```

```
      data,
      v => v.length,
      d => d.Destination
    );

    const words = Array.from(destinationCounts, ([text, size]) => ({ text,
size }));

    const fontScale = d3.scaleLinear()
      .domain([d3.min(words, d => d.size), d3.max(words, d => d.size)])
      .range([10, 60]);

    d3.layout.cloud()
      .size([width, height])
      .words(words)
      .padding(5)
      .fontSize(d => fontScale(d.size))
      .on("end", draw)
      .start();

    function draw(words) {
      svg.append("g")
        .attr("transform", `translate(${width / 2}, ${height / 2})`)
        .selectAll("text")
```

```
        .data(words)

        .enter().append("text")

        .style("font-size", d => `${d.size}px`)

        .style("fill", () => d3.schemeCategory10[Math.floor(Math.random()
* 10)])

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

        .attr("transform", d => `translate(${d.x}, ${d.y})
rotate(${d.rotate})`)

        .text(d => d.text);

    }

  });
```

9] Violin Plot



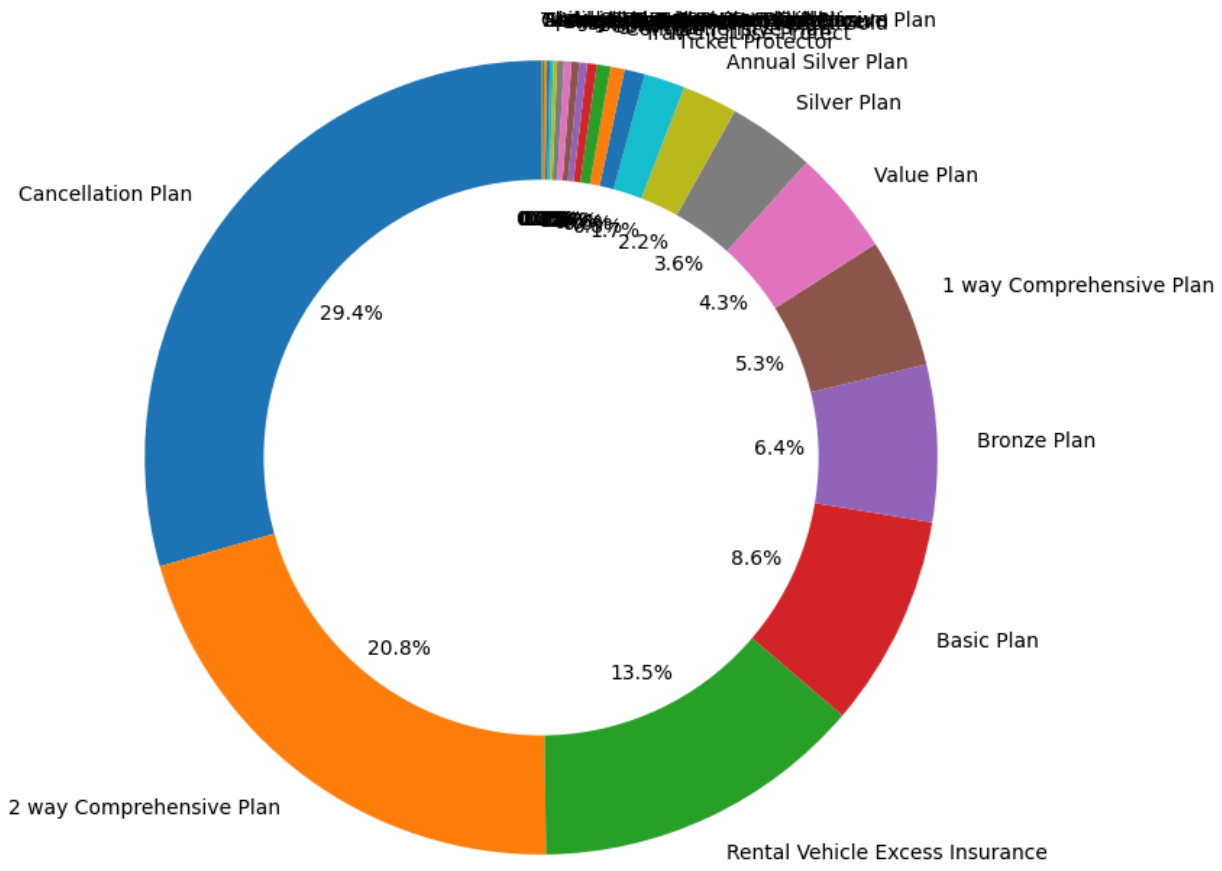

```
d3.csv("travel_insurance.csv").then(data => {
```

```
    const svg = d3.select("svg");

    svg.selectAll("*").remove();



    const width = +svg.attr("width");

    const height = +svg.attr("height");

    const margin = { top: 20, right: 30, bottom: 100, left: 50 };



    const productNames = Array.from(new Set(data.map(d => d["Product
Name"])));



    const xScale = d3.scaleBand()

      .domain(productNames)

      .range([margin.left, width - margin.right])

      .padding(0.5);



    const yScale = d3.scaleLinear()

      .domain([d3.min(data, d => +d["Net Sales"]), d3.max(data, d =>
+d["Net Sales"])])

      .range([height - margin.bottom, margin.top]);



    const yAxis = d3.axisLeft(yScale);

    const xAxis = d3.axisBottom(xScale);



    svg.append("g")
```

```
    .attr("transform", `translate(${margin.left}, 0)`)
    .call(yAxis);


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


// Group data by product name
const groupedData = d3.group(data, d => d["Product Name"]);


groupedData.forEach((values, productName) => {
  const binGenerator = d3.bin()
    .domain(yScale.domain())
    .thresholds(10)
    .value(d => +d["Net Sales"]);


  const bins = binGenerator(values);


  const maxBinLength = d3.max(bins, b => b.length);
```

```
    const xViolinScale = d3.scaleLinear()
      .domain([-maxBinLength, maxBinLength])
      .range([0, xScale.bandwidth()]);

    const areaGenerator = d3.area()
      .x0(d => xViolinScale(-d.length))
      .x1(d => xViolinScale(d.length))
      .y(d => yScale(d.x0))
      .curve(d3.curveCatmullRom);

    svg.append("path")
      .datum(bins)
      .attr("transform", `translate(${xScale(productName)},0)`)
      .attr("fill", "#69b3a2")
      .attr("opacity", 0.6)
      .attr("stroke", "black")
      .attr("stroke-width", 1)
      .attr("d", areaGenerator);
  });
});
```

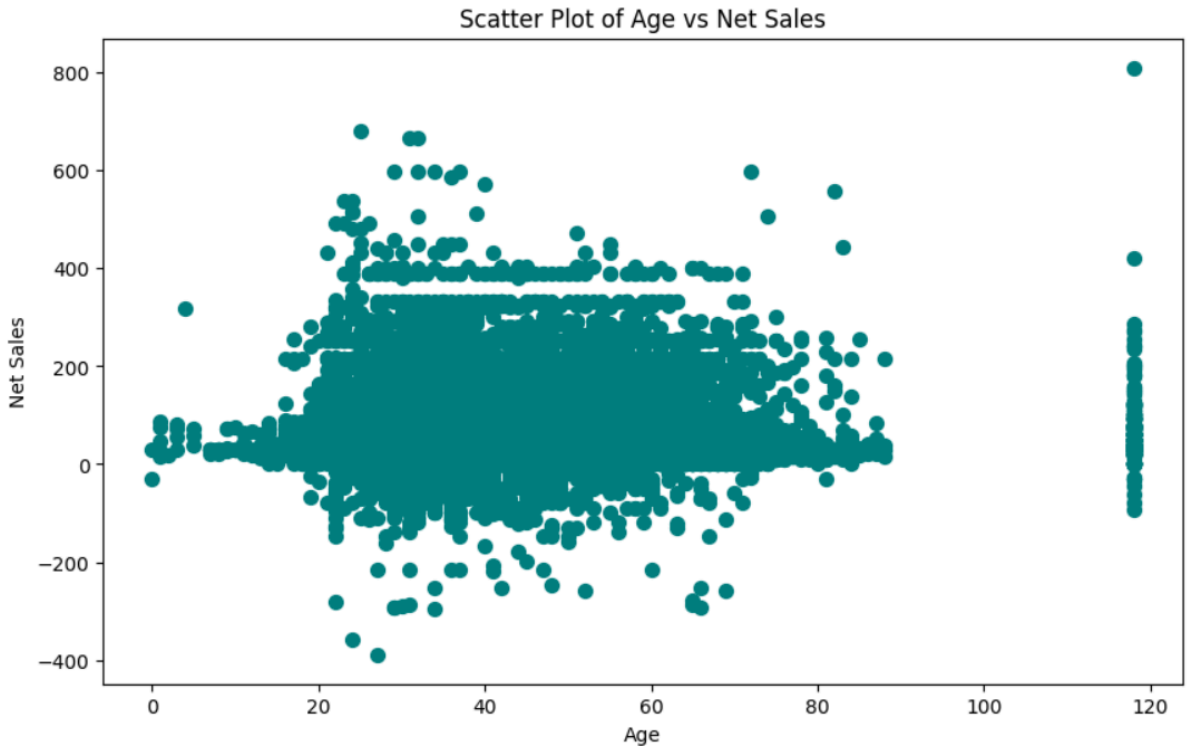
10] Pie Chart




```
d3.csv("travel_insurance.csv").then(data => {

    const svg = d3.select("svg");

    svg.selectAll("*").remove(); // Clear previous chart



    const width = +svg.attr("width");

    const height = +svg.attr("height");

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

```
  const group = svg.append("g")
    .attr("transform", `translate(${width / 2}, ${height / 2})`);

  const pie = d3.pie()
    .value(d => 1)
    .sort(null);

  const arc = d3.arc()
    .innerRadius(0)
    .outerRadius(radius);

  const color = d3.scaleOrdinal(d3.schemeCategory10);

  group.selectAll("path")
    .data(pie(data))
    .enter()
    .append("path")
    .attr("d", arc)
    .attr("fill", (d, i) => color(i));
});
```

11] Scatter Plot




```
d3.csv("travel_insurance.csv").then(data => {

    const svg = d3.select("svg");

    svg.selectAll("*").remove(); // Clear previous chart



    const width = +svg.attr("width");

    const height = +svg.attr("height");

    const margin = { top: 20, right: 20, bottom: 40, left: 40 };



    const xScale = d3.scaleLinear()

      .domain([0, d3.max(data, d => +d.Age)])

      .range([margin.left, width - margin.right]);



    const yScale = d3.scaleLinear()
```

```
      .domain([d3.min(data, d => +d["Net Sales"]), d3.max(data, d =>
+d["Net Sales"])])

      .range([height - margin.bottom, margin.top]);



    svg.append("g")

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

      .call(d3.axisBottom(xScale));



    svg.append("g")

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

      .call(d3.axisLeft(yScale));



    svg.selectAll("circle")

      .data(data)

      .enter()

      .append("circle")

      .attr("cx", d => xScale(+d.Age))

      .attr("cy", d => yScale(+d["Net Sales"]))

      .attr("r", 5)

      .attr("fill", "teal");

  });
```

Output and Observation : -

```
Pearson Correlation Coefficient: 0.03775601148380212

P-value: 2.0125848439927503e-21

Reject the null hypothesis: There is a significant correlation
```

In our analysis of the dataset, the Pearson correlation coefficient between Age and Net Sales was calculated to be approximately 0.0378. This value indicates a very weak positive correlation between the two variables. However, despite the weak correlation, the p-value associated with this correlation was extremely low, approximately 2.01e-21.

Given that the p-value is significantly less than the conventional alpha level of 0.05, we reject the null hypothesis. This suggests that there is indeed a statistically significant correlation between Age and Net Sales in the dataset.

While the correlation is weak, the significant p-value indicates that changes in Age do have a statistically meaningful relationship with Net Sales. This finding could imply that as customers' ages change, their purchasing behavior reflected in Net Sales might also be influenced, albeit subtly. It suggests that further exploration of other factors or more complex relationships in the dataset could provide deeper insights into customer behavior and sales patterns.

Conclusion :- From this experiment, I learned to plot advance visualization like 3d chart , box and whisker plot , violin plot etc in d3 js.