BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering

Course – Advanced Data Visualization (ADV)

Name	Tejal Subhash Komb
UID	2021600037
Batch	A
Lab no	07

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.

Dataset :-

Month	WeekOfM	DayOfWe	Make	AccidentA	DayOfWe	MonthCla	WeekOfM	Sex	MaritalSta	Age	Fault	PolicyType	VehicleCat	VehiclePri	FraudFoun	PolicyNum	RepNumbe	Deductible	DriverRati	Days_Polic	Days_Polic	PastNumb	Ag
Dec	5	Wednesda	Honda	Urban	Tuesday	Jan	1	Female	Single	21	Policy Hol	Sport - Lia	Sport	more than	0	1	12	300	1	more than	more than	none	3
Jan	3	Wednesda	Honda	Urban	Monday	Jan	4	Male	Single	34	Policy Hol	Sport - Col	Sport	more than	0	2	15	400	4	more than	more than	none	6
Oct	5	Friday	Honda	Urban	Thursday	Nov	2	Male	Married	47	Policy Hol	Sport - Col	Sport	more than	0	3	7	400	3	more than	more than	1	7
Jun	2	Saturday	Toyota	Rural	Friday	Jul	1	Male	Married	65	Third Party	Sedan - Lia	Sport	20000 to 2	0	4	4	400	2	more than	more than	1	m
Jan	5	Monday	Honda	Urban	Tuesday	Feb	2	Female	Single	27	Third Party	Sport - Col	Sport	more than	0	5	3	400	1	more than	more than	none	5
Oct	4	Friday	Honda	Urban	Wednesda	Nov	1	Male	Single	20	Third Party	Sport - Col	Sport	more than	0	6	12	400	3	more than	more than	none	5
Feb	1	Saturday	Honda	Urban	Monday	Feb	3	Male	Married	36	Third Party	Sport - Col	Sport	more than	0	7	14	400	1	more than	more than	1	7
Nov	1	Friday	Honda	Urban	Tuesday	Mar	4	Male	Single	0	Policy Hol	Sport - Col	Sport	more than	0	8	1	400	4	more than	more than	1	ne
Dec	4	Saturday	Honda	Urban	Wednesda	Dec	5	Male	Single	30	Policy Hol	Sport - Col	Sport	more than	0	9	7	400	4	more than	more than	none	6
Apr	3	Tuesday	Ford	Urban	Wednesda	Apr	3	Male	Married	42	Policy Hol	Utility - All	Utility	more than	0	10	7	400	1	more than	more than	2 to 4	m
Mar	2	Sunday	Mazda	Urban	Wednesda	Mar	3	Male	Single	71	Policy Hol	Sedan - All	Sedan	more than	0	11	7	400	3	more than	more than	none	m
Mar	5	Monday	Honda	Urban	Monday	Mar	5	Male	Married	52	Policy Hol	Sedan - Lia	Sport	20000 to 2	0	12	13	400	1	more than	more than	2 to 4	m
Jan	3	Friday	Ford	Urban	Friday	Jan	3	Male	Married	28	Policy Hol	Sedan - Lia	Sport	more than	0	13	11	400	1	more than	more than	1	7
Jan	5	Friday	Honda	Rural	Wednesda	Feb	1	Male	Single	0	Third Party	Sedan - Co	Sedan	more than	0	14	12	400	3	more than	more than	none	ne
Jan	5	Monday	Ford	Urban	Thursday	Feb	1	Male	Married	61	Policy Hol	Sedan - Lia	Sport	more than	0	15	3	400	1	more than	more than	none	m
Aug	4	Tuesday	Ford	Urban	Monday	Aug	5	Male	Single	38	Policy Hol	Sedan - Lia	Sport	more than	0	16	16	400	1	more than	more than	none	6
Apr	4	Thursday	Ford	Urban	Wednesda	May	1	Male	Married	41	Policy Hol	Sedan - All	Sedan	more than	0	17	15	400	4	more than	more than	none	7
Jul	5	Sunday	Chevrolet	Urban	Wednesda	Aug	1	Female	Married	28	Third Party	Sedan - Co	Sedan	20000 to 2	0	18	6	400	1	more than	more than	none	7
May	4	Thursday	Pontiac	Urban	Monday	May	5	Male	Single	32	Policy Hol	Sedan - Lia	Sport	20000 to 2	0	19	6	400	1	more than	more than	1	7
Apr	4	Monday	Honda	Urban	Tuesday	May	1	Male	Married	30	Third Party	Sedan - Lia	Sport	more than	0	20	2	400	2	more than	more than	2 to 4	6
Apr	2	Friday	Mazda	Urban	Tuesday	May	1	Male	Married	40	Policy Hol	Sedan - Lia	Sport	20000 to 2	0	21	3	400	1	more than	more than	1	m
Jan	2	Saturday	Chevrolet	Urban	Monday	Jan	2	Male	Married	47	Policy Hol	Sedan - Co	Sedan	20000 to 2	0	22	13	400	2	more than	more than	1	m
Aug	3	Sunday	Mazda	Urban	Thursday	Aug	5	Male	Married	63	Policy Hol	Sedan - Lia	Sport	20000 to 2	0	23	8	400	3	more than	more than	1	m
Jun	3	Saturday	Pontiac	Urban	Tuesday	Jun	3	Male	Single	31	Third Party	Sedan - Lia	Sport	30000 to 3	0	24	5	400	3	more than	more than	none	6
Sep	3	Friday	Mazda	Urban	Friday	Sep	3	Male	Married	45	Policy Hol	Sedan - All	Sedan	more than	0	25	12	400	3	more than	more than	more than	m
Mar	3	Monday	Pontiac	Urban	Tuesday	Apr	1	Male	Married	60	Policy Hol	Sedan - Lia	Sport	20000 to 2	0	26	16	400	4	more than	more than	more than	m
Mar	3	Thursday	Honda	Urban	Thursday	Jun	4	Male	Married	21	Policy Hol	Sedan - Co	Sedan	30000 to 3	0	27	1	400	2	more than	more than	more than	5
May	3	Sunday	Accura	Urban	Friday	May	4	Male	Married	42	Policy Hol	Sedan - All	Sedan	30000 to 3	0	28	1	400	3	more than	more than	2 to 4	7

Dataset description :-

- **Month, WeekOfMonth, DayOfWeek**: Date of the accident.

- **Make**: The manufacturer of the vehicle (e.g., Honda, Toyota).
- **AccidentArea**: Whether the accident occurred in an urban or rural area.
- **DayOfWeekClaimed, MonthClaimed, WeekOfMonthClaimed**: Date when the claim was filed.
- **Sex, MaritalStatus, Age**: Demographic information of the policyholder.
- **Fault**: Whether the policyholder was at fault in the accident.
- **PolicyType, VehicleCategory, VehiclePrice**: Details about the insurance policy and the vehicle.
- **FraudFound_P**: Indicator of whether the claim was found to be fraudulent.
- **PolicyNumber, RepNumber, Deductible, DriverRating**: Policy and claim-specific information.
- **Days_Policy_Accident, Days_Policy_Claim**: Days since the policy started to the accident and claim.
- **PastNumberOfClaims**: Number of past claims made by the policyholder.
- **AgeOfVehicle, AgeOfPolicyHolder**: Age of the vehicle and policyholder.
- **PoliceReportFiled, WitnessPresent**: Whether a police report was filed and if a witness was present.
- **AgentType**: Type of agent handling the claim.
- **NumberOfSuppliments**: Number of supplementary documents submitted.
- **AddressChange_Claim**: Whether there was an address change at the time of the claim.
- **NumberOfCars**: Number of cars insured under the policy.
- **Year**: Year the vehicle was manufactured.
- **BasePolicy**: Type of base policy (e.g., Liability, Collision, All Perils).

BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering

Implementation :- D3.js implementation

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Insurance Data Visualization</title>
    <script src="https://d3js.org/d3.v6.min.js"></script>
    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/d3-
cloud/1.2.5/d3.layout.cloud.min.js"></script>
</head>
<body>

    <h1>Insurance Data Visualization</h1>

    <h2>Number of Claims by Car Make (Bar Chart)</h2>
    <svg id="barChart" width="800" height="500"></svg>

    <h2>Policy Type Distribution with Percentages (Pie Chart)</h2>
    <svg id="pieChart" width="800" height="800"></svg>

    <h2>Distribution of Age of Policy Holders (Histogram)</h2>
    <svg id="histogramChart" width="800" height="600"></svg>

    <h2>Number of Claims by Month (Line Chart)</h2>
    <svg id="lineChart" width="800" height="500"></svg>
```

```
    <h2>Age Distribution by Policy Type (Box Plot)</h2>
    <svg id="boxPlotChart" width="800" height="600"></svg>

    <h2>3D Scatter Plot of Age, Policy Type, and Vehicle Price</h2>
    <div id="3dPlot" style="width:800px; height:600px;"></div>

    <h2>Age Distribution (Violin Plot)</h2>
    <div id="violinPlot" style="width:800px; height:600px;"></div>

    <h2>Vehicle Category Word Cloud</h2>
    <svg id="wordCloud" width="900" height="700"></svg>

    <script>
        // Bar Chart Code
        const barSvg = d3.select("#barChart"),
              barMargin = { top: 50, right: 60, bottom: 60, left: 60 },
              barChartWidth = 800 - barMargin.left - barMargin.right,
              barChartHeight = 500 - barMargin.top - barMargin.bottom;

        const barChart = barSvg.append("g")
                               .attr("transform",
`translate(${barMargin.left},${barMargin.top})`);

        d3.csv("./fraud_oracle.csv").then(function(data) {
            const claimsByMake = d3.rollups(data, v => v.length, d => d.Make);
            const x = d3.scaleBand()
                        .domain(claimsByMake.map(d => d[0]))
                        .range([0, barChartWidth])
                        .padding(0.3);
```

```
const y = d3.scaleLinear()
            .domain([0, d3.max(claimsByMake, d => d[1])])
            .nice()
            .range([barChartHeight, 0]);

const colorScale = d3.scaleOrdinal()
                     .domain(claimsByMake.map(d => d[0]))
                     .range(d3.schemeCategory10);

barChart.selectAll("rect")
        .data(claimsByMake)
        .enter()
        .append("rect")
        .attr("x", d => x(d[0]))
        .attr("y", d => y(d[1]))
        .attr("width", x.bandwidth())
        .attr("height", d => barChartHeight - y(d[1]))
        .attr("fill", d => colorScale(d[0]));

barChart.append("g")
        .attr("transform", `translate(0, ${barChartHeight})`)
        .call(d3.axisBottom(x))
        .selectAll("text")
        .attr("transform", "rotate(-45)")
        .style("text-anchor", "end");

barChart.append("g")
        .call(d3.axisLeft(y));
```

```
            barChart.append("text")
                    .attr("x", barChartWidth / 2)
                    .attr("y", -20)
                    .attr("text-anchor", "middle")
                    .style("font-size", "18px")
                    .style("font-weight", "bold")
                    .text("Number of Claims by Car Make");
        });

        // Pie Chart Code
        const pieSvg = d3.select("#pieChart")
                        .append("g")
                        .attr("transform", `translate(400,400)`); // Center within the
800x800 SVG
        const color = d3.scaleOrdinal(d3.schemeSet3);

        d3.csv("./fraud_oracle.csv").then(function(data) {
            const policyTypeCounts = d3.rollups(data, v => v.length, d => d.PolicyType);
            const totalCount = d3.sum(policyTypeCounts, d => d[1]);
            const pie = d3.pie().value(d => d[1]);
            const arc = d3.arc().innerRadius(0).outerRadius(250); // Adjusted outer radius
for better fit
            const outerArc = d3.arc().innerRadius(300).outerRadius(300);

            pieSvg.selectAll("path")
                  .data(pie(policyTypeCounts))
                  .enter()
                  .append("path")
                  .attr("d", arc)
```

```
                .attr("fill", d => color(d.data[0]));

            pieSvg.selectAll("text")
                .data(pie(policyTypeCounts))
                .enter()
                .append("text")
                .attr("transform", d => `translate(${outerArc.centroid(d)})`)
                .attr("dy", "0.35em")
                .style("text-anchor", d => (d.endAngle + d.startAngle) / 2 > Math.PI ?
"end" : "start")
                .text(d => `${d.data[0]} (${((d.data[1] / totalCount) *
100).toFixed(2)}%)`);

            pieSvg.append("text")
                .attr("x", 0)
                .attr("y", -280)
                .attr("text-anchor", "middle")
                .style("font-size", "18px")
                .style("font-weight", "bold")
                .text("Policy Type Distribution with Percentages");
        });

        // Histogram Code
        const histogramSvg = d3.select("#histogramChart"),
            histogramMargin = { top: 50, right: 50, bottom: 70, left: 70 },
            histogramWidth = 800 - histogramMargin.left - histogramMargin.right,
            histogramHeight = 600 - histogramMargin.top - histogramMargin.bottom;

        const histogramG = histogramSvg.append("g")
```

```
                                    .attr("transform",
`translate(${histogramMargin.left},${histogramMargin.top})`);

        d3.csv("./fraud_oracle.csv").then(function(data) {
            const ageData = data.map(d => +d.Age); // Convert Age to a numeric value
            const x = d3.scaleLinear()
                        .domain([0, d3.max(ageData)]) // Automatically scale with age data
                        .range([0, histogramWidth]);

            const histogram = d3.histogram()
                                .value(d => d)
                                .domain(x.domain())
                                .thresholds(x.ticks(20)); // Adjust the number of bins as
needed

            const bins = histogram(ageData);

            const y = d3.scaleLinear()
                        .domain([0, d3.max(bins, d => d.length)])
                        .nice()
                        .range([histogramHeight, 0]);

            histogramG.selectAll("rect")
                      .data(bins)
                      .enter()
                      .append("rect")
                      .attr("x", d => x(d.x0) + 1)
                      .attr("y", d => y(d.length))
                      .attr("width", d => x(d.x1) - x(d.x0) - 1)
```

```
                    .attr("height", d => histogramHeight - y(d.length))
                    .attr("fill", "steelblue");

          histogramG.append("g")
                    .attr("transform", `translate(0,${histogramHeight})`)
                    .call(d3.axisBottom(x));

          histogramG.append("g")
                    .call(d3.axisLeft(y));

          histogramG.append("text")
                    .attr("x", histogramWidth / 2)
                    .attr("y", histogramHeight + histogramMargin.bottom - 13)
                    .style("text-anchor", "middle")
                    .text("Age of Policy Holders");

          histogramG.append("text")
                    .attr("transform", "rotate(-90)")
                    .attr("x", -histogramHeight / 2)
                    .attr("y", -histogramMargin.left + 15)
                    .style("text-anchor", "middle")
                    .text("Number of Claims");

          histogramSvg.append("text")
                      .attr("x", (histogramWidth + histogramMargin.left +
histogramMargin.right) / 2)
                      .attr("y", histogramMargin.top / 2)
                      .style("text-anchor", "middle")
                      .style("font-size", "18px")
```

```
                        .style("font-weight", "bold")
                        .text("Distribution of Age of Policy Holders");
        });

        // Line Chart Code (Number of Claims by Month)
        const lineSvg = d3.select("#lineChart"),
              lineMargin = { top: 50, right: 50, bottom: 60, left: 60 },
              lineWidth = 800 - lineMargin.left - lineMargin.right,
              lineHeight = 500 - lineMargin.top - lineMargin.bottom;

        const lineChart = lineSvg.append("g")
                                 .attr("transform",
`translate(${lineMargin.left},${lineMargin.top})`);

        d3.csv("./fraud_oracle.csv").then(function(data) {
            const claimsByMonth = d3.rollups(data, v => v.length, d => d.MonthClaimed);
            const x = d3.scaleBand().domain(claimsByMonth.map(d => d[0])).range([0,
lineWidth]).padding(0.3);
            const y = d3.scaleLinear().domain([0, d3.max(claimsByMonth, d =>
d[1])]).range([lineHeight, 0]);

            const line = d3.line()
                           .x(d => x(d[0]) + x.bandwidth() / 2)
                           .y(d => y(d[1]));

            lineChart.append("path")
                     .datum(claimsByMonth)
                     .attr("fill", "none")
                     .attr("stroke", "orange")
```

```
                .attr("stroke-width", 2)
                .attr("d", line);

        lineChart.append("g")
                .attr("transform", `translate(0,${lineHeight})`)
                .call(d3.axisBottom(x));

        lineChart.append("g")
                .call(d3.axisLeft(y));

        lineChart.append("text")
                .attr("x", lineWidth / 2)
                .attr("y", -20)
                .attr("text-anchor", "middle")
                .style("font-size", "18px")
                .style("font-weight", "bold")
                .text("Number of Claims by Month");
    });

    // Box Plot Code for Age by Policy Type
    const boxSvg = d3.select("#boxPlotChart"),
          boxMargin = { top: 50, right: 50, bottom: 60, left: 60 },
          boxWidth = 800 - boxMargin.left - boxMargin.right,
          boxHeight = 600 - boxMargin.top - boxMargin.bottom;

    const boxChart = boxSvg.append("g")
                           .attr("transform",
`translate(${boxMargin.left},${boxMargin.top})`);
```

```
        d3.csv("./fraud_oracle.csv").then(function(data) {
            const ageByPolicyType = d3.groups(data, d => d.PolicyType).map(([key, values])
=> ({
                key,
                values: values.map(d => +d.Age)
            }));
            const x = d3.scaleBand().domain(ageByPolicyType.map(d => d.key)).range([0,
boxWidth]).padding(0.2);
            const y = d3.scaleLinear().domain([0, d3.max(ageByPolicyType, d =>
d3.max(d.values))]).range([boxHeight, 0]);

            boxChart.append("g").call(d3.axisLeft(y));
            boxChart.append("g").attr("transform",
`translate(0,${boxHeight})`).call(d3.axisBottom(x));

            const colorBoxScale =
d3.scaleOrdinal(d3.schemeTableau10).domain(ageByPolicyType.map(d => d.key));

            ageByPolicyType.forEach(group => {
                const q1 = d3.quantile(group.values.sort(d3.ascending), 0.25);
                const median = d3.quantile(group.values, 0.5);
                const q3 = d3.quantile(group.values, 0.75);
                const interQuantileRange = q3 - q1;
                const min = d3.max([0, q1 - 1.5 * interQuantileRange]);
                const max = d3.min([d3.max(y.domain()), q3 + 1.5 * interQuantileRange]);

                boxChart.append("line")
                        .attr("x1", x(group.key) + x.bandwidth() / 2)
                        .attr("x2", x(group.key) + x.bandwidth() / 2)
```

```
                .attr("y1", y(min))
                .attr("y2", y(max))
                .attr("stroke", "black");

        boxChart.append("rect")
                .attr("x", x(group.key))
                .attr("y", y(q3))
                .attr("width", x.bandwidth())
                .attr("height", y(q1) - y(q3))
                .attr("fill", colorBoxScale(group.key));

        boxChart.append("line")
                .attr("x1", x(group.key))
                .attr("x2", x(group.key) + x.bandwidth())
                .attr("y1", y(median))
                .attr("y2", y(median))
                .attr("stroke", "black");
    });

    boxSvg.append("text")
          .attr("x", (boxWidth + boxMargin.left + boxMargin.right) / 2)
          .attr("y", boxMargin.top / 2)
          .style("text-anchor", "middle")
          .style("font-size", "18px")
          .style("font-weight", "bold")
          .text("Age Distribution by Policy Type");
});

// 3D Scatter Plot Code
```

BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering

```
        Plotly.d3.csv("./fraud_oracle.csv", function(data) {  // Adjusted file path to
"./fraud_oracle.csv"
            const ages = data.map(d => +d.Age);
            const policyTypes = data.map(d => d.PolicyType);
            const vehiclePrices = data.map(d => +d.VehiclePrice.replace(/[^\\d.-]/g, ''));

            const trace = {
                x: ages,
                y: policyTypes,
                z: vehiclePrices,
                mode: 'markers',
                marker: {
                    size: 5,
                    color: ages,
                    colorscale: 'Viridis'
                },
                type: 'scatter3d'
            };

            const layout = {
                title: "3D Chart of Age, Policy Type, and Vehicle Price",
                scene: {
                    xaxis: { title: 'Age' },
                    yaxis: { title: 'Policy Type', tickangle: -90 },
                    zaxis: { title: 'Vehicle Price' }
                },
                margin: {
                    l: 100,
                    r: 100,
```

```
                b: 50,
                t: 50
            }
        };

        Plotly.newPlot('3dPlot', [trace], layout);
    });

    // Violin Plot Code
    Plotly.d3.csv("./fraud_oracle.csv", function(data) {  // Adjusted file path to
"./fraud_oracle.csv"
        const ages = data.map(d => +d.Age);
        const trace = {
            y: ages,
            type: 'violin',
            box: {
                visible: true
            },
            meanline: {
                visible: true
            }
        };

        const layout = {
            title: "Age Distribution (Violin Plot)",
            yaxis: {
                title: "Age"
            }
        };
```

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

        // Word Cloud Code
        d3.csv("./fraud_oracle.csv").then(function(data) {  // Adjusted file path to
"./fraud_oracle.csv"
            const vehicleCategoryField = "VehicleCategory";
            const vehicleCategoryCounts = d3.rollup(
                data, v => v.length, d => d[vehicleCategoryField]
            );

            const words = Array.from(vehicleCategoryCounts, ([text, size]) => ({ text, size
}));
            console.log("Word cloud data: ", words);

            words.forEach(word => {
                word.size = Math.max(word.size * 0.005, 8);
            });

            const width = 900, height = 700;
            const svg = d3.select("#wordCloud")
                          .attr("width", width)
                          .attr("height", height);

            const layout = d3.layout.cloud()
                .size([width, height])
                .words(words)
                .padding(5)
```

```
            .rotate(() => ~~(Math.random() * 2) * 90)
            .fontSize(d => d.size)
            .on("end", draw);

        layout.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 => `${d.size}px`)
                .style("fill", (d, i) => d3.schemeCategory10[i % 10])
                .attr("text-anchor", "middle")
                .attr("transform", d => `translate(${d.x},${d.y}) rotate(${d.rotate})`)
                .text(d => d.text);
        }
    });
  </script>
</body>
</html>
```

BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

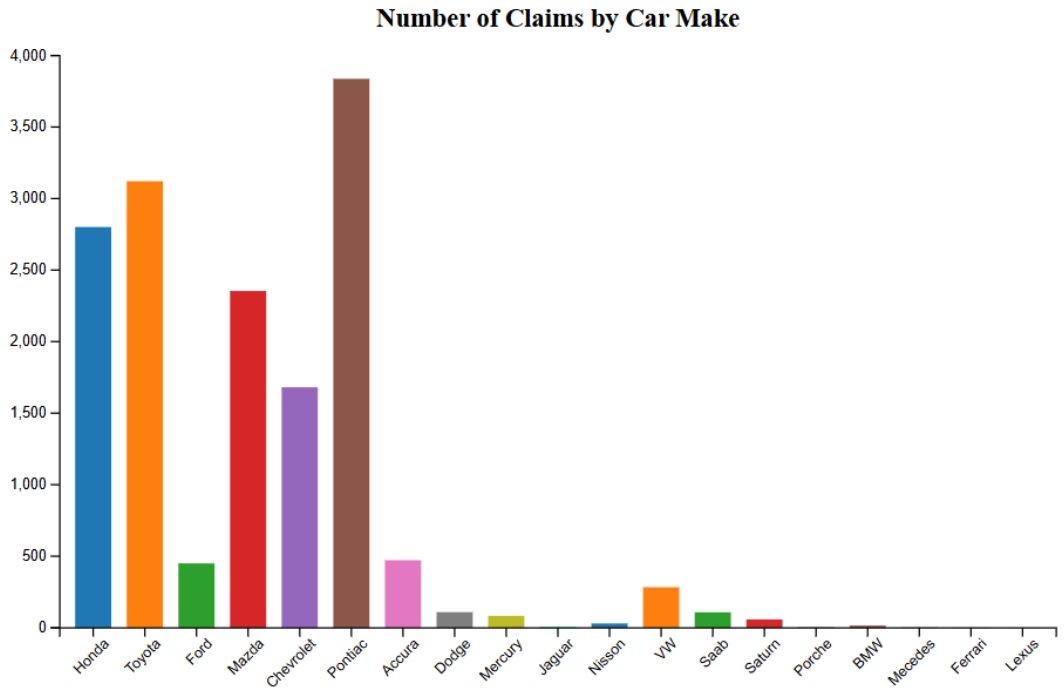
Department of Computer Science and Engineering

Output :-

1] Bar Chart :-

Insurance Data Visualization

Number of Claims by Car Make (Bar Chart)

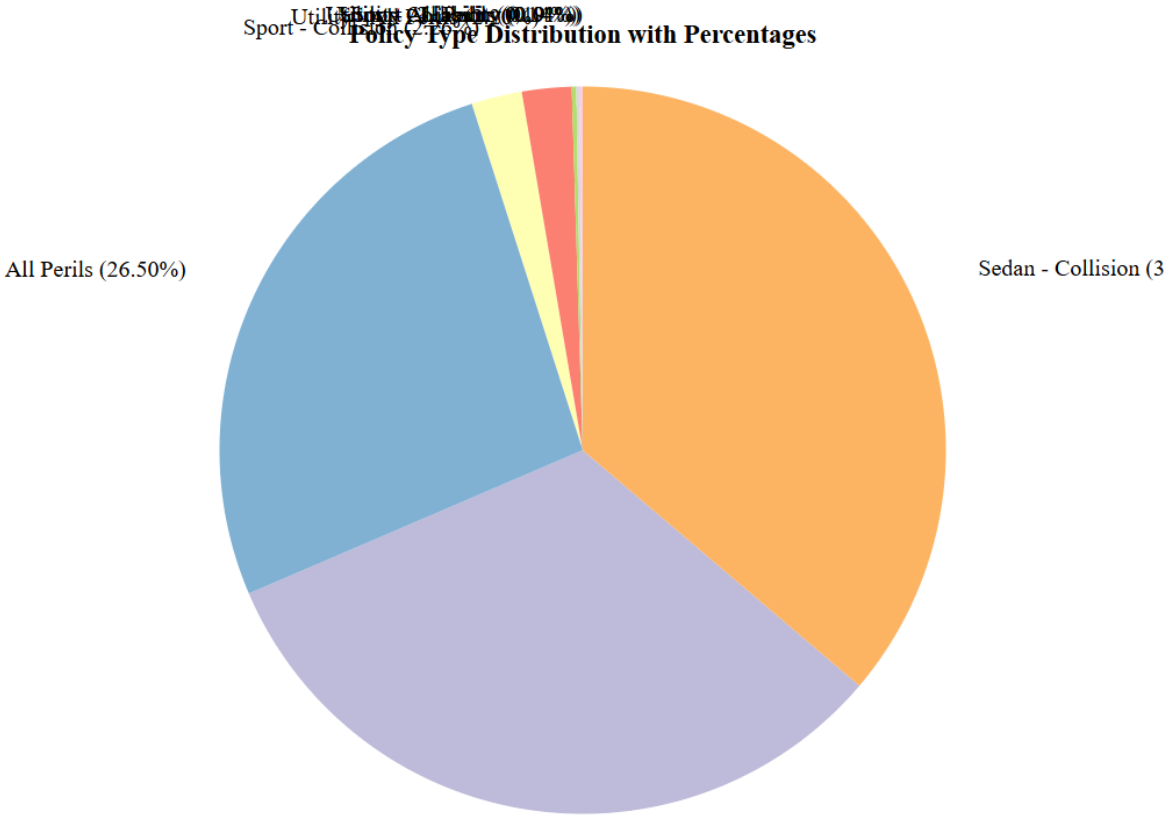


BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering

2] pie chart :-

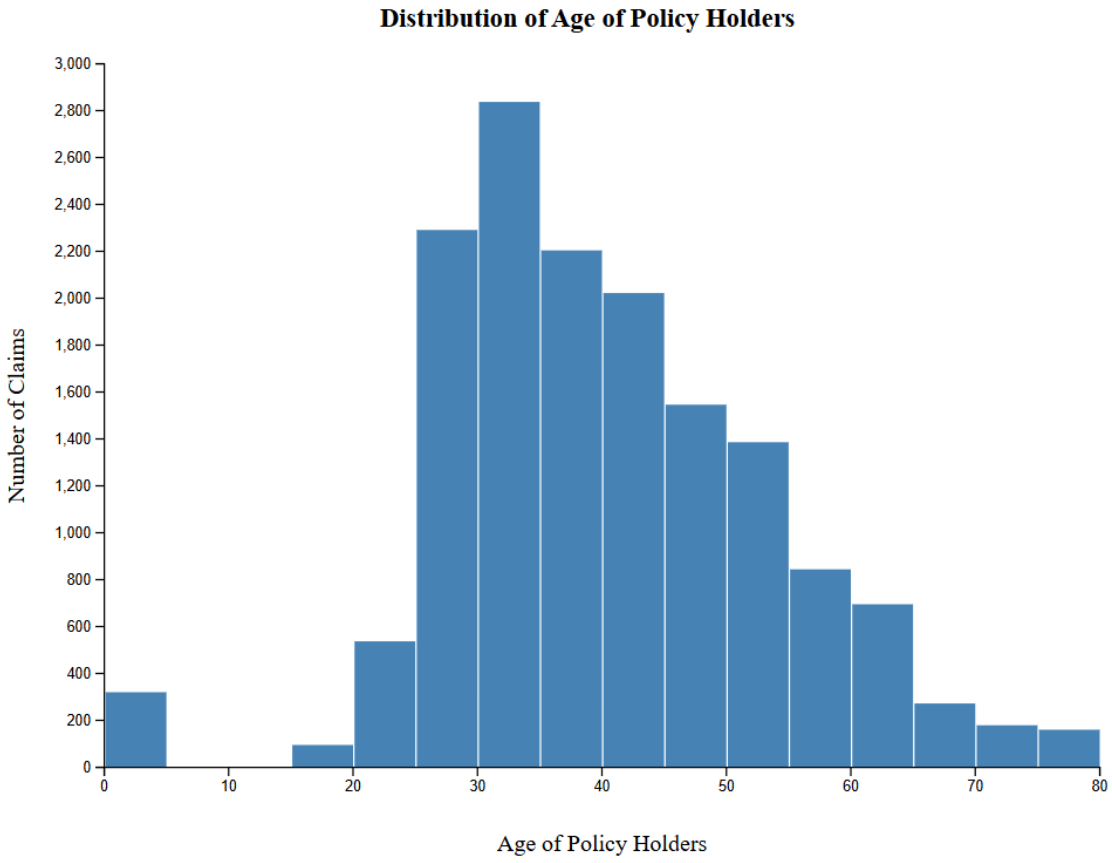
Policy Type Distribution with Percentages (Pie Chart)



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering

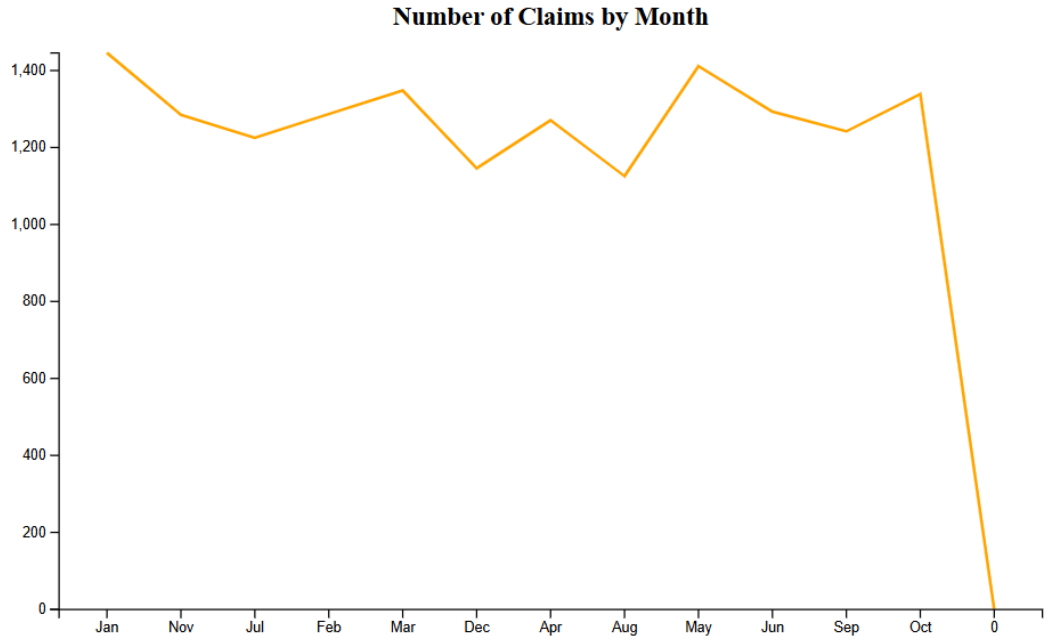
Distribution of Age of Policy Holders (Histogram)



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering

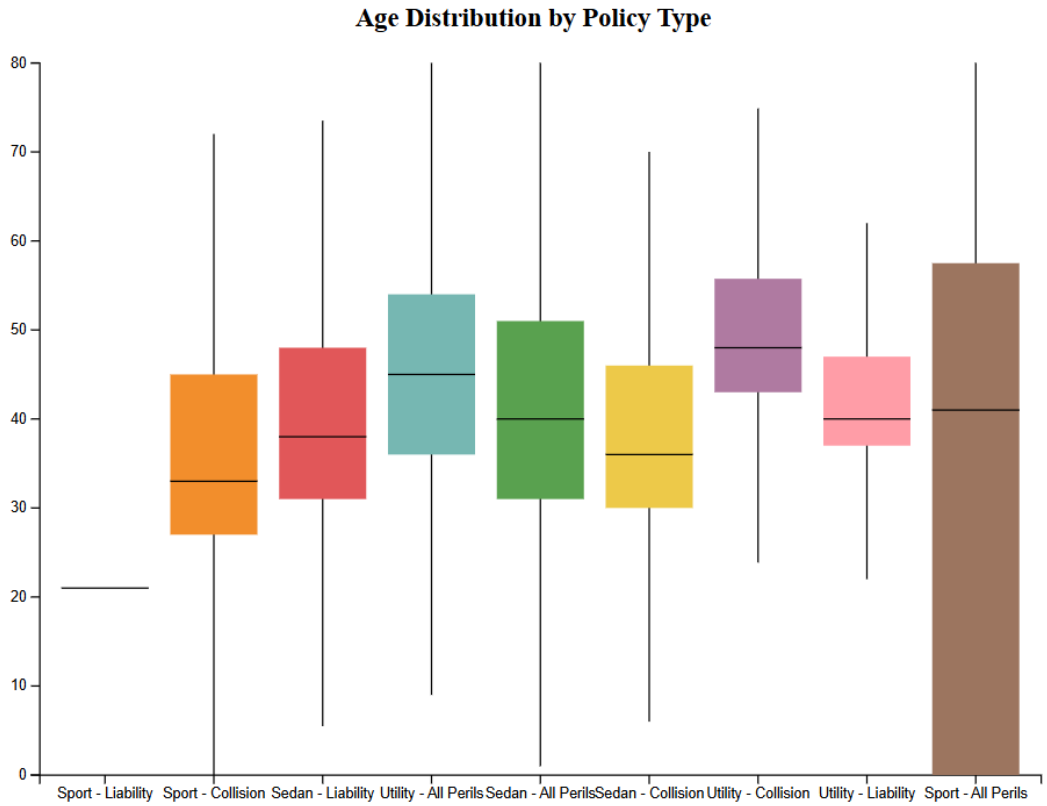
Number of Claims by Month (Line Chart)



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering

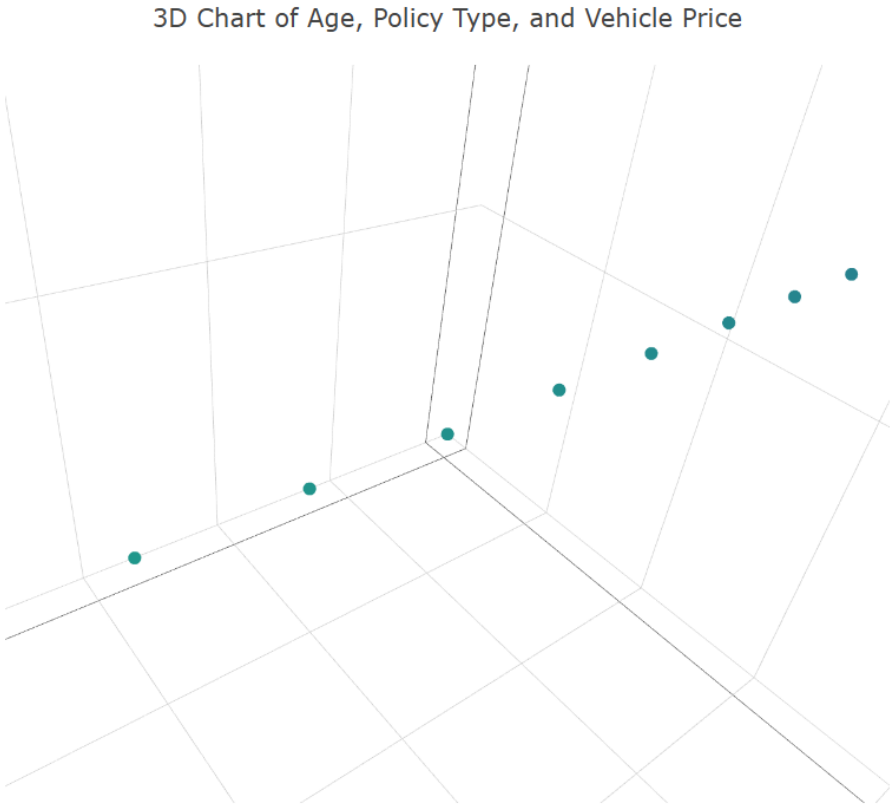
Age Distribution by Policy Type (Box Plot)



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering

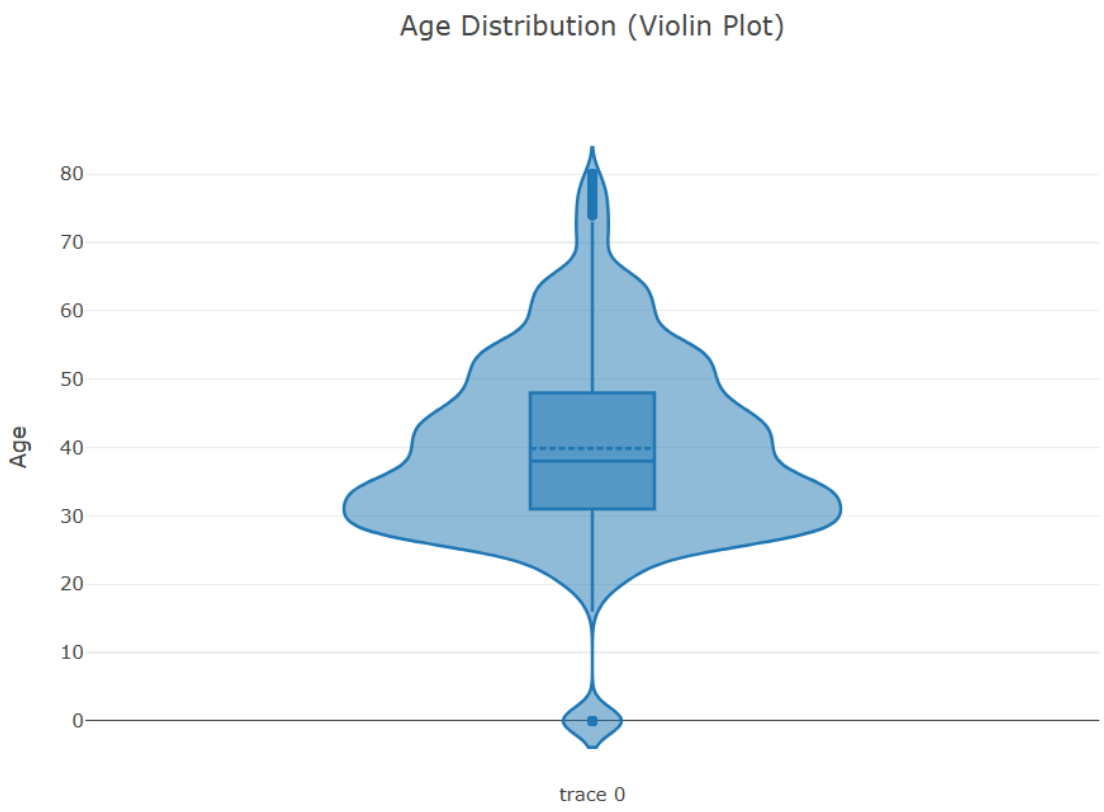
3D Scatter Plot of Age, Policy Type, and Vehicle Price



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering

Age Distribution (Violin Plot)



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering

Vehicle Category Word Cloud

BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400093-India
(Autonomous College Affiliated to University of Mumbai)

Department of Computer Science and Engineering