NAME:	Sakshi Bisen
UID No.	2021600008
BRANCH:	B.E CSE-AIML
BATCH:	I
SUBJECT	Advanced Data Visualization
EXPERIMENT No.	7
DATE:	13/10/2024

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

DATASET: https://www.kaggle.com/datasets/shivamb/vehicle-claim-fraud-detection

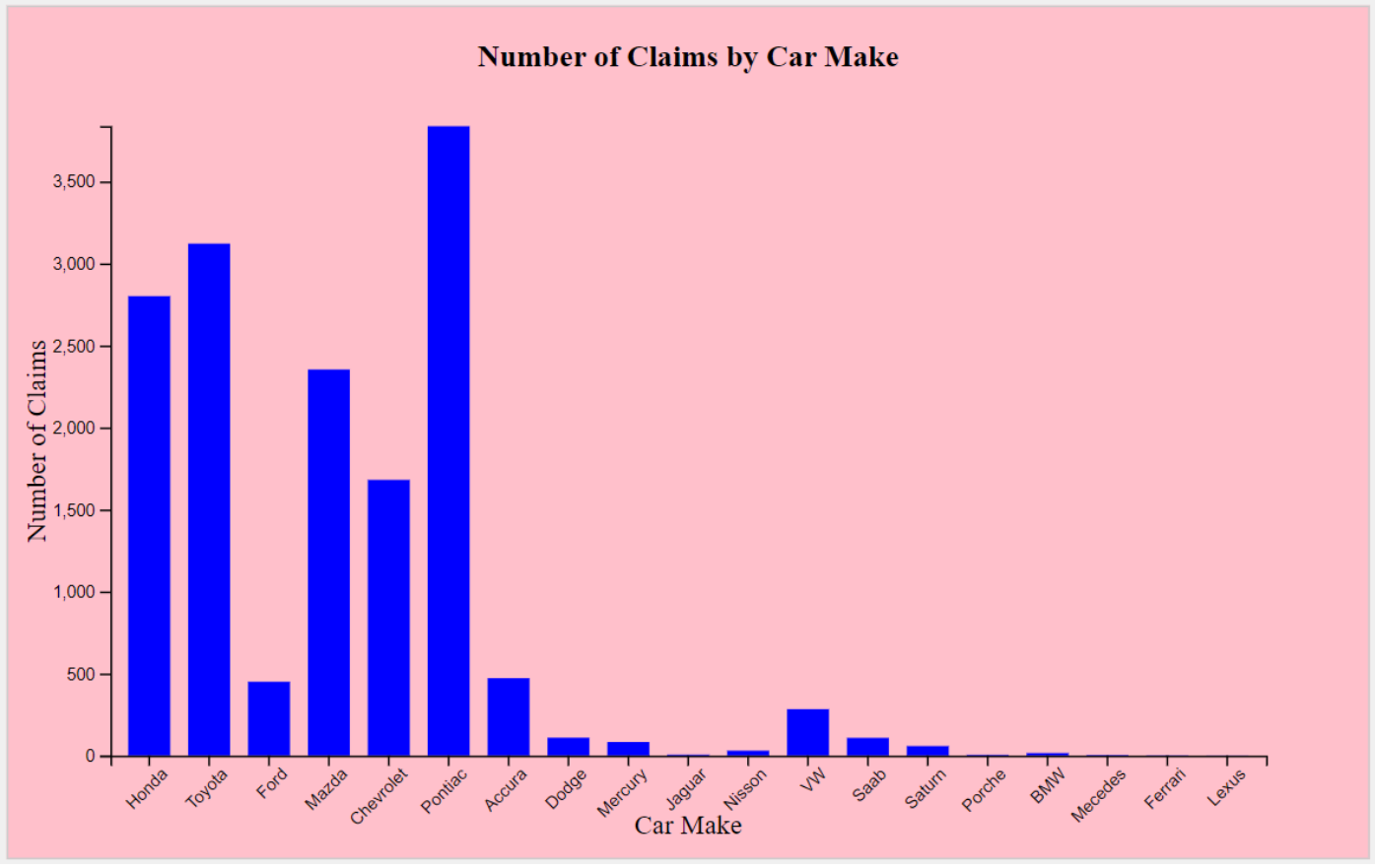
The dataset used here contains details about insurance claims, including columns like Age, Sex, VehiclePrice, and PolicyType, among others, with the aim of detecting potential fraud. It provides valuable insights into patterns of claims based on customer demographics, vehicle characteristics, and policy details.

ANALYSIS:

1] Bar Graph

```
d3.csv("../fraud_oracle.csv").then(function(data) {
    const svg = d3.select("svg"),
          margin = { top: 70, right: 60, bottom: 60, left: 60 },
          width = svg.attr("width") - margin.left - margin.right,
          height = svg.attr("height") - margin.top - margin.bottom;
    const chart = svg.append("g")
                     .attr("transform", `translate(${margin.left},${margin.top})`);

    const claimsByMake = d3.rollups(data, v => v.length, d => d.Make);

    const x = d3.scaleBand()
                .domain(claimsByMake.map(d => d[0]))
                .range([0, width])
                .padding(0.3);

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

    chart.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 => height - y(d[1]))
        .attr("fill", "blue");

    chart.append("g")
        .attr("transform", "translate(0," + height + ")")
        .call(d3.axisBottom(x))
        .selectAll("text")
        .attr("transform", "rotate(-45)")
        .style("text-anchor", "end");
```

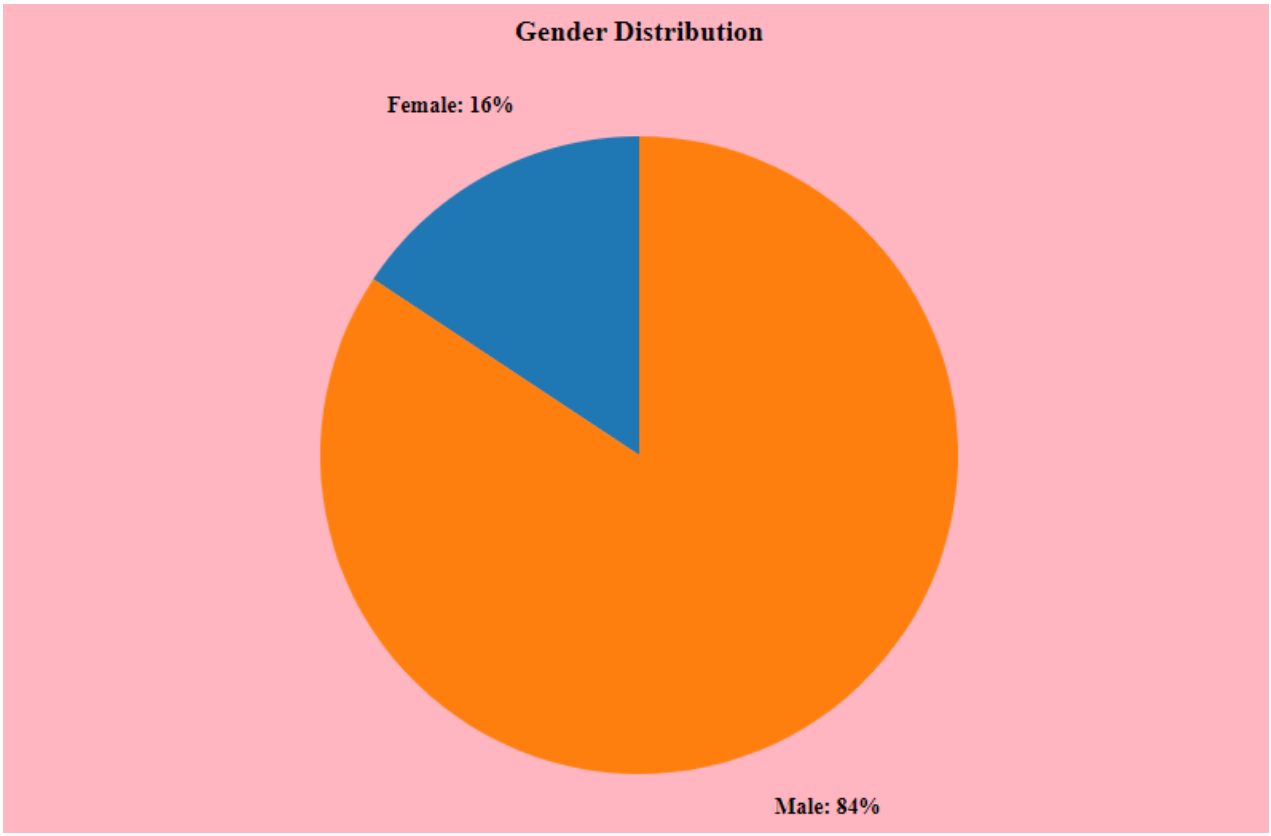
```
    chart.append("g")
        .call(d3.axisLeft(y));

    chart.append("text")
        .attr("transform", "rotate(-90)")
        .attr("x", -height / 2)
        .attr("y", -margin.left / 1.1)
        .attr("dy", "1em")
        .style("text-anchor", "middle")
        .text("Number of Claims");

    chart.append("text")
        .attr("x", width / 2)
        .attr("y", height + margin.bottom / 1.3)
        .style("text-anchor", "middle")
        .text("Car Make");

    svg.append("text")
        .attr("x", (width + margin.left + margin.right) / 2)
        .attr("y", margin.top / 2)
        .attr("text-anchor", "middle")
        .style("font-size", "18px")
        .style("font-weight", "bold")
        .text("Number of Claims by Car Make");
```



The above Bar Graph shows the Number of Policy Claims as per the Car Model(Make). The above graph shows that the car make Pontiac has resulted to higher number of policy claims.

2] Pie Graph

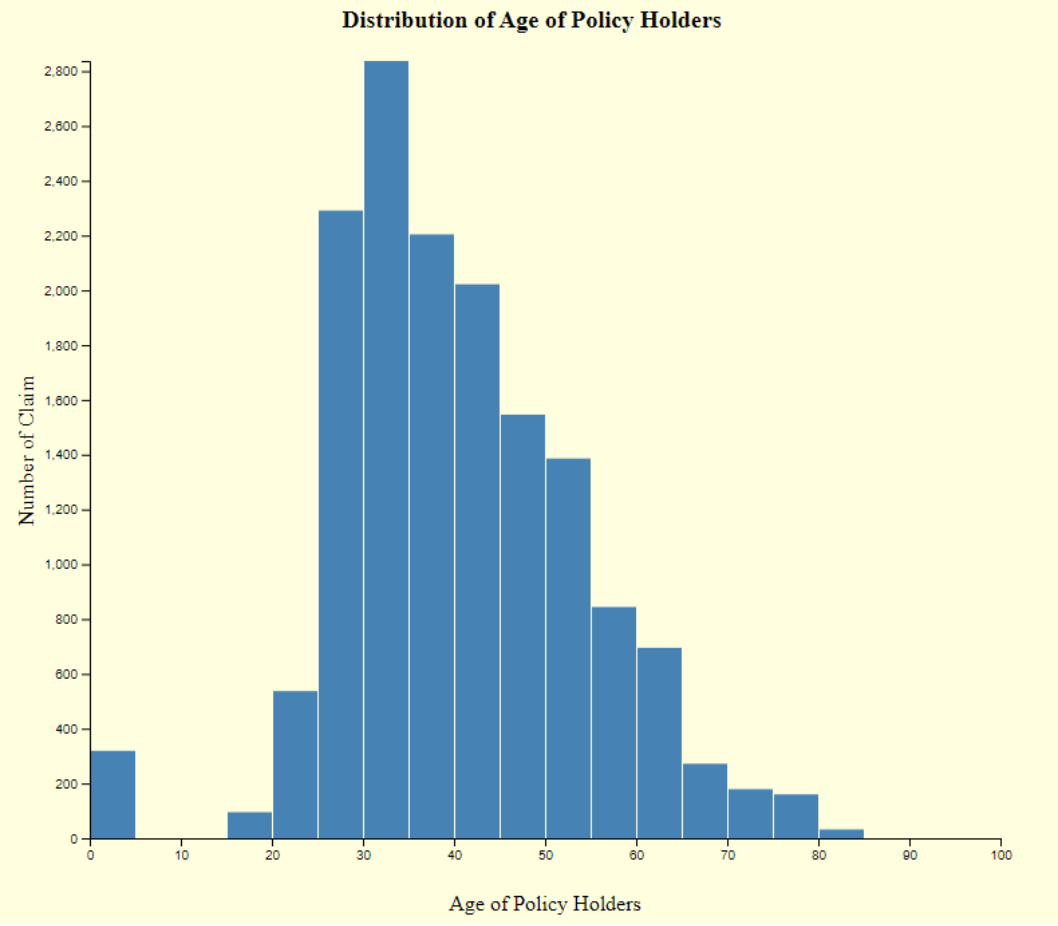
```
const width = 800, height = 800, radius = Math.min(width, height) / 3;
const svg = d3.select("svg")
              .append("g")
              .attr("transform", `translate(${width / 2},${height / 2})`);
const color = d3.scaleOrdinal(d3.schemeCategory10);

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(radius);
    const outerArc = d3.arc().innerRadius(radius * 1.1).outerRadius(radius * 1.1);

    svg.selectAll("path")
        .data(pie(policyTypeCounts)).enter()
        .append("path").attr("d", arc)
        .attr("fill", d => color(d.data[0]));

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

    svg.append("text")
        .attr("x", 0).attr("y", -radius - 50)
        .attr("text-anchor", "middle")
        .style("font-size", "18px")
        .style("font-weight", "bold")
        .text("Policy Type Distribution with Percentages");
});
```



The above pie chart shows the distribution of policy claims according to genders. The chart states that males drivers have a higher number of claims than female.

3] Histogram

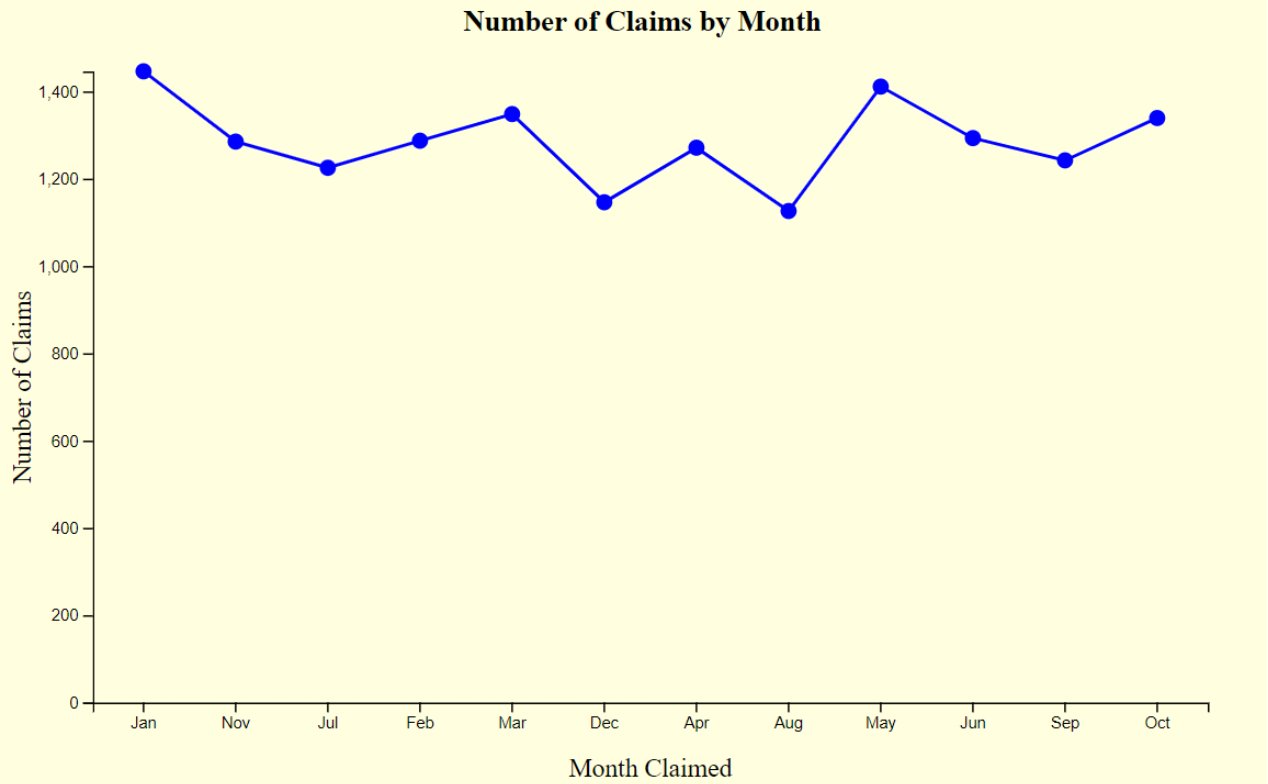
```
d3.csv("../fraud_oracle.csv").then(function(data) {
    const svg = d3.select("svg"),margin = { top: 50, right: 50, bottom: 70, left: 70 },
          width = svg.attr("width") - margin.left - margin.right,
          height = svg.attr("height") - margin.top - margin.bottom;

    const x = d3.scaleLinear().domain([0, 100]).range([0, width]);
    const y = d3.scaleLinear().range([height, 0]);
    const bins = d3.histogram().value(d => d.Age).domain(x.domain()).thresholds(x.ticks(20));
    const histogramData = bins(data);
    y.domain([0, d3.max(histogramData, d => d.length)]);
    const g = svg.append("g").attr("transform", `translate(${margin.left},${margin.top})`);

    g.selectAll("rect")
        .data(histogramData).enter().append("rect").attr("x", 1)
        .attr("transform", d => `translate(${x(d.x0)},${y(d.length)})`)
        .attr("width", d => x(d.x1) - x(d.x0) - 1)
        .attr("height", d => height - y(d.length))
        .attr("fill", "steelblue");

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

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

    g.append("text").attr("x", width / 2).attr("y", height + margin.bottom / 1.3)
        .style("text-anchor", "middle").text("Age of Policy Holders");

    g.append("text").attr("transform", "rotate(-90)").attr("x", -height / 2)
        .attr("y", -margin.left / 1.2).attr("dy", "1em")
        .style("text-anchor", "middle").text("Number of Claim");

    svg.append("text").attr("x", (width + margin.left + margin.right) / 2)
        .attr("y", margin.top / 2).attr("text-anchor", "middle")
        .style("font-size", "18px").style("font-weight", "bold")
        .text("Distribution of Age of Policy Holders");
});
```



The above Histogram shows the distributions of policy claims as per age. The median comes to around 38 and it has the highest number of claims.

4] Line Graph

```
d3.csv("../fraud_oracle.csv").then(function(data) {
    const svg = d3.select("svg"),
          margin = { top: 50, right: 50, bottom: 60, left: 60 },
          width = svg.attr("width") - margin.left - margin.right,
          height = svg.attr("height") - margin.top - margin.bottom;
    const chart = svg.append("g")
                     .attr("transform", `translate(${margin.left},${margin.top})`);

    const claimsByMonth = d3.rollups(data, v => v.length, d => d.MonthClaimed);
    const x = d3.scaleBand().domain(claimsByMonth.map(d => d[0])).range([0, width]).padding(0.1);
    const y = d3.scaleLinear().domain([0, d3.max(claimsByMonth, d => d[1])]).range([height, 0]);
    const line = d3.line().x(d => x(d[0]) + x.bandwidth() / 2).y(d => y(d[1]));

    chart.append("path").datum(claimsByMonth).attr("fill", "none")
         .attr("stroke", "blue").attr("stroke-width", 2).attr("d", line);

    chart.selectAll("circle").data(claimsByMonth).enter()
         .append("circle").attr("cx", d => x(d[0]) + x.bandwidth() / 2)
         .attr("cy", d => y(d[1])).attr("r", 5).attr("fill", "blue");

    chart.append("g").attr("transform", "translate(0," + height + ")").call(d3.axisBottom(x));

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

    chart.append("text").attr("transform", "rotate(-90)").attr("x", -height / 2)
        .attr("y", -margin.left / 1.1).attr("dy", "1em")
        .style("text-anchor", "middle").text("Number of Claims");

    chart.append("text").attr("x", width / 2).attr("y", height + margin.bottom / 1.3)
        .style("text-anchor", "middle").text("Month Claimed");

    svg.append("text").attr("x", (width + margin.left + margin.right) / 2)
        .attr("y", margin.top / 2).attr("text-anchor", "middle").style("font-size", "18px")
        .style("font-weight", "bold").text("Number of Claims by Month");
});
```



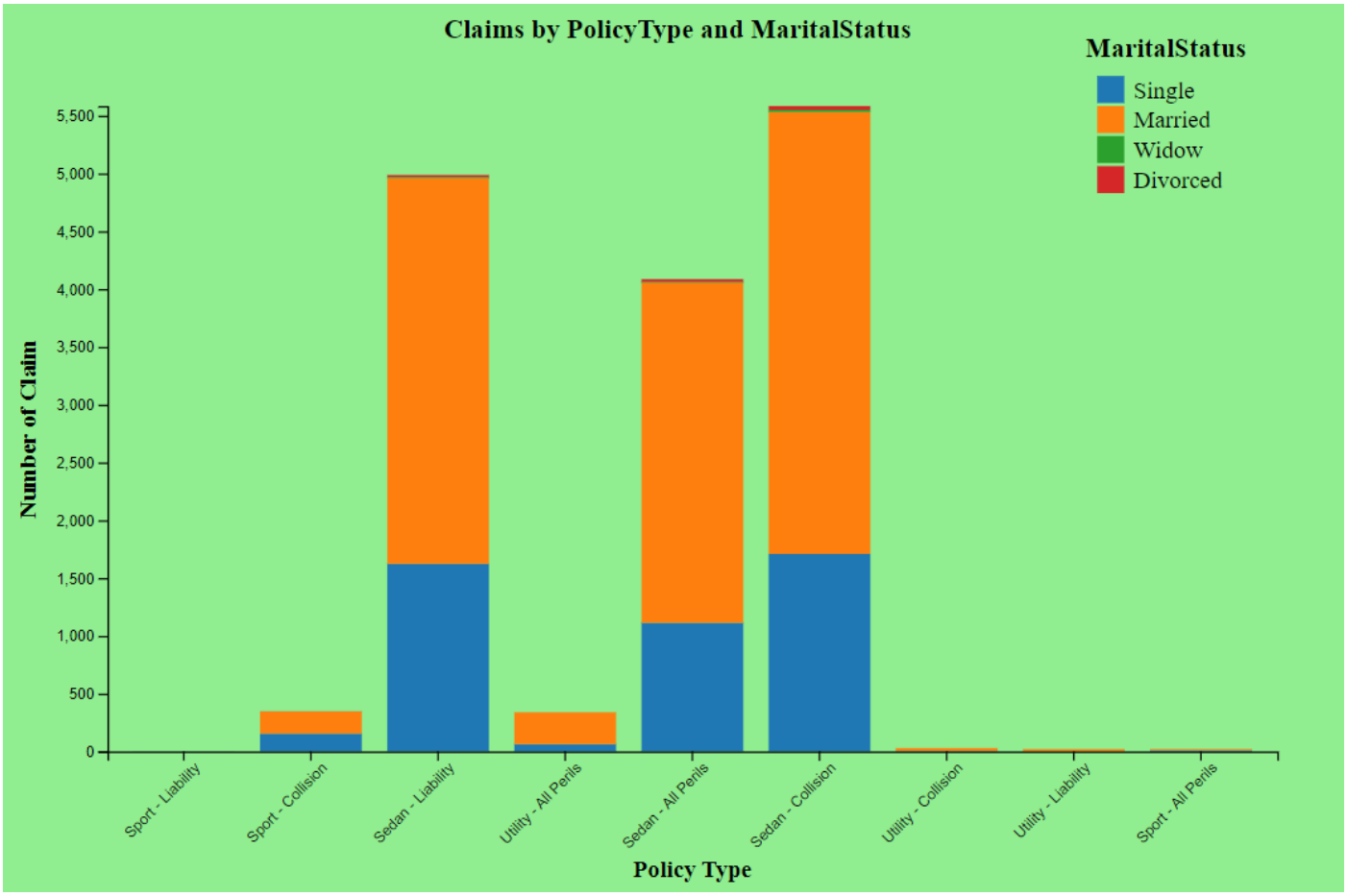
The above Timeline Graph shows the number of claims as per the months. The number is close for all the months but is seen to be higher in the month of January and May.

5] Stacked Bar Graph

```
d3.csv("../fraud_oracle.csv").then(function(data) {
    const svg = d3.select("svg"),
          margin = { top: 70, right: 50, bottom: 100, left: 70 },
          width = svg.attr("width") - margin.left - margin.right,
          height = svg.attr("height") - margin.top - margin.bottom;

    const chart = svg.append("g").attr("transform", `translate(${margin.left},${margin.top})`);
    const dataGrouped = d3.rollups(data, v => v.length, d => d.PolicyType, d => d.MaritalStatus);
    const flatData = [];
    dataGrouped.forEach(([policyType, maritalStatuses]) => {
        const record = { PolicyType: policyType };
        maritalStatuses.forEach(([maritalStatus, count]) => {
            record[maritalStatus] = count;
        });
        flatData.push(record);
    });

    svg.append("text")
       .attr("x", (width + margin.left + margin.right) /1.16)
       .attr("y", margin.top / 1.9)
       .attr("text-anchor", "middle")
       .style("font-size", "18px")
       .style("font-weight", "bold")
       .text("MaritalStatus");

    const keys = Array.from(new Set(data.map(d => d.MaritalStatus)));
    const x = d3.scaleBand()
                .domain(flatData.map(d => d.PolicyType))
                .range([0, width])
                .padding([0.2]);

    const y = d3.scaleLinear()
                .domain([0, d3.max(flatData, d => d3.sum(keys, k => d[k] || 0))])
                .range([height, 0]);
```

```
const color = d3.scaleOrdinal().domain(keys).range(d3.schemeCategory10);

const stackedData = d3.stack().keys(keys)(flatData);

chart.selectAll("g").data(stackedData).enter().append("g")
      .attr("fill", d => color(d.key)).selectAll("rect").data(d => d)
      .enter().append("rect").attr("x", d => x(d.data.PolicyType))
      .attr("y", d => y(d[1])).attr("height", d => y(d[0]) - y(d[1]))
      .attr("width", x.bandwidth());

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

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

const legend = svg.append("g")
                  .attr("transform", `translate(${width + margin.right - 100}, 50)`);

keys.forEach((key, i) => {
    legend.append("rect")
          .attr("x", 0).attr("y", i * 20).attr("width", 18)
          .attr("height", 18).style("fill", color(key));

    legend.append("text")
          .attr("x", 24).attr("y", i * 20 + 9)
          .attr("dy", "0.35em").text(key);
});
```

```
    chart.append("text")
        .attr("transform", "rotate(-90)")
        .attr("x", -height / 2)
        .attr("y", -margin.left / 1.1)
        .attr("dy", "1em")
        .style("text-anchor", "middle")
        .style("font-weight", "bold")
        .text("Number of Claim");

    chart.append("text")
        .attr("x", width / 2)
        .attr("y", height + margin.bottom / 1.2)
        .style("text-anchor", "middle")
        .style("font-weight", "bold")
        .text("Policy Type");

    svg.append("text")
       .attr("x", (width + margin.left + margin.right) / 2)
       .attr("y", margin.top / 2 - 10)
       .attr("text-anchor", "middle")
       .style("font-size", "18px")
       .style("font-weight", "bold")
       .text("Claims by PolicyType and MaritalStatus");
});
```

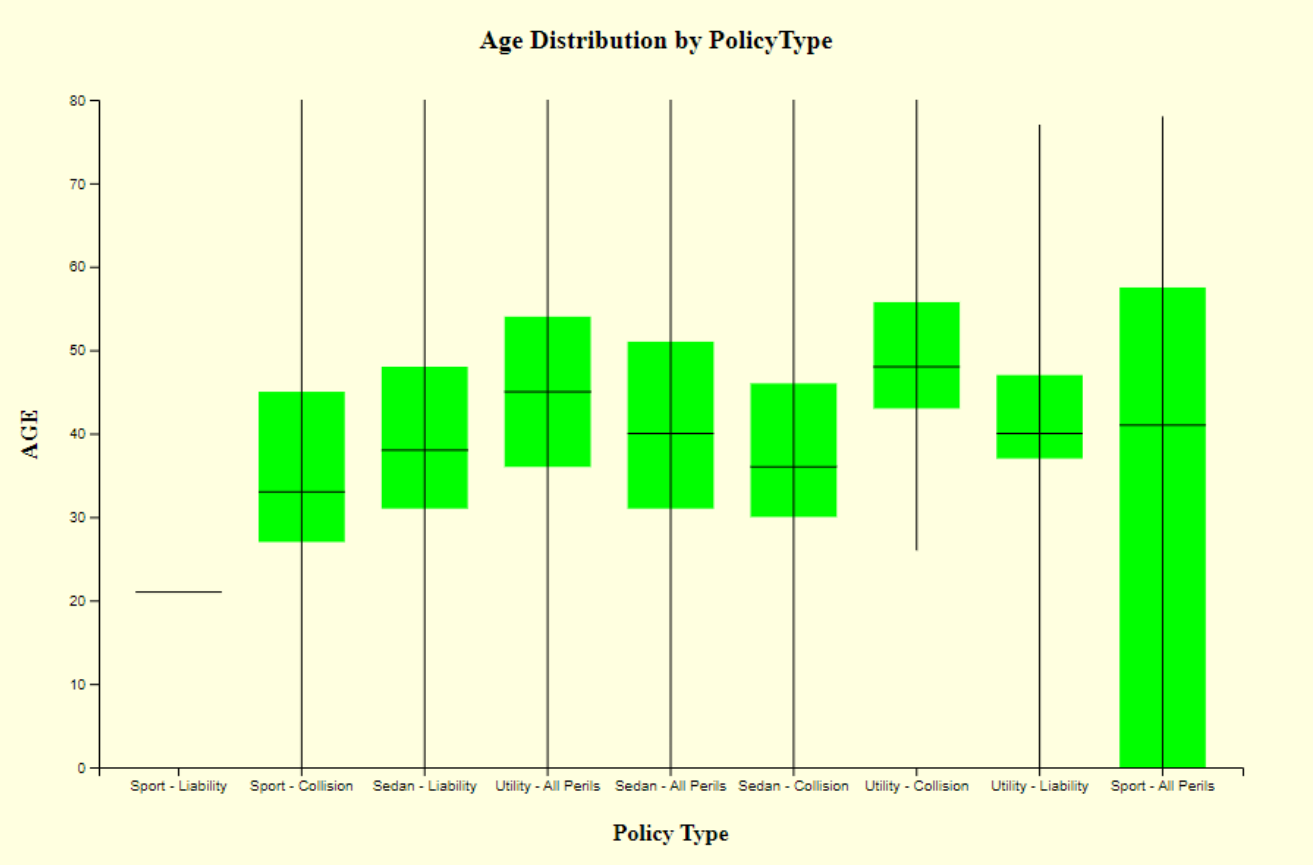


The above Staked Bar Graph is shows us the distribution of number of claims as per policy type and marital status. The above graph shows that the maximum number of claims are filed by married people and the least by widows. The most common claim policy types are Sedan – Liability and Sedan – Collision.

6] Box and Wisker Graph

```
d3.csv("../fraud_oracle.csv").then(function(data) {
    const svg = d3.select("svg"),
          margin = { top: 70, right: 50, bottom: 75, left: 70 },
          width = svg.attr("width") - margin.left - margin.right,
          height = svg.attr("height") - margin.top - margin.bottom;
    const chart = svg.append("g")
                     .attr("transform", `translate(${margin.left},${margin.top})`);

    const policyData = d3.group(data, d => d.PolicyType);
    const summary = Array.from(policyData, ([key, values]) => {
        const ages = values.map(d => +d.Age).sort(d3.ascending);
        const q1 = d3.quantile(ages, 0.25);
        const median = d3.quantile(ages, 0.5);
        const q3 = d3.quantile(ages, 0.75);
        const min = d3.min(ages);
        const max = d3.max(ages);
        return { key, q1, median, q3, min, max };
    });

    const x = d3.scaleBand().domain(summary.map(d => d.key)).range([0, width]).padding([0.3]);

    const y = d3.scaleLinear().domain([0, d3.max(summary, d => d.max)]).range([height, 0]);

    chart.selectAll("rect").data(summary).enter().append("rect").attr("x", d => x(d.key))
         .attr("y", d => y(d.q3)).attr("height", d => y(d.q1) - y(d.q3))
         .attr("width", x.bandwidth()).attr("fill", "lime");

    chart.selectAll("line.median").data(summary).enter().append("line")
         .attr("class", "median").attr("x1", d => x(d.key))
         .attr("x2", d => x(d.key) + x.bandwidth()).attr("y1", d => y(d.median))
         .attr("y2", d => y(d.median)).attr("stroke", "black");
```

```
    chart.selectAll("line.whisker").data(summary).enter().append("line")
         .attr("class", "whisker").attr("x1", d => x(d.key) + x.bandwidth() / 2)
         .attr("x2", d => x(d.key) + x.bandwidth() / 2).attr("y1", d => y(d.min))
         .attr("y2", d => y(d.max)).attr("stroke", "black");

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

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

    chart.append("text").attr("transform", "rotate(-90)")
        .attr("x", -height / 2).attr("y", -margin.left / 1.2)
        .attr("dy", "1em").style("text-anchor", "middle")
        .style("font-weight", "bold").text("AGE");

    chart.append("text").attr("x", width / 2)
        .attr("y", height + margin.bottom / 1.5).style("text-anchor", "middle")
        .style("font-weight", "bold").text("Policy Type");

    svg.append("text").attr("x", (width + margin.left + margin.right) / 2)
       .attr("y", margin.top / 2 ).attr("text-anchor", "middle")
       .style("font-size", "18px").style("font-weight", "bold")
       .text("Age Distribution by PolicyType");
});
```



The above Box and Wisker Graph shows distribution of policy claims as per policy type and age. It can be seen that Sport – All Perils has group of people from almost all the ages and Sport – Liability does not have that much crowd.

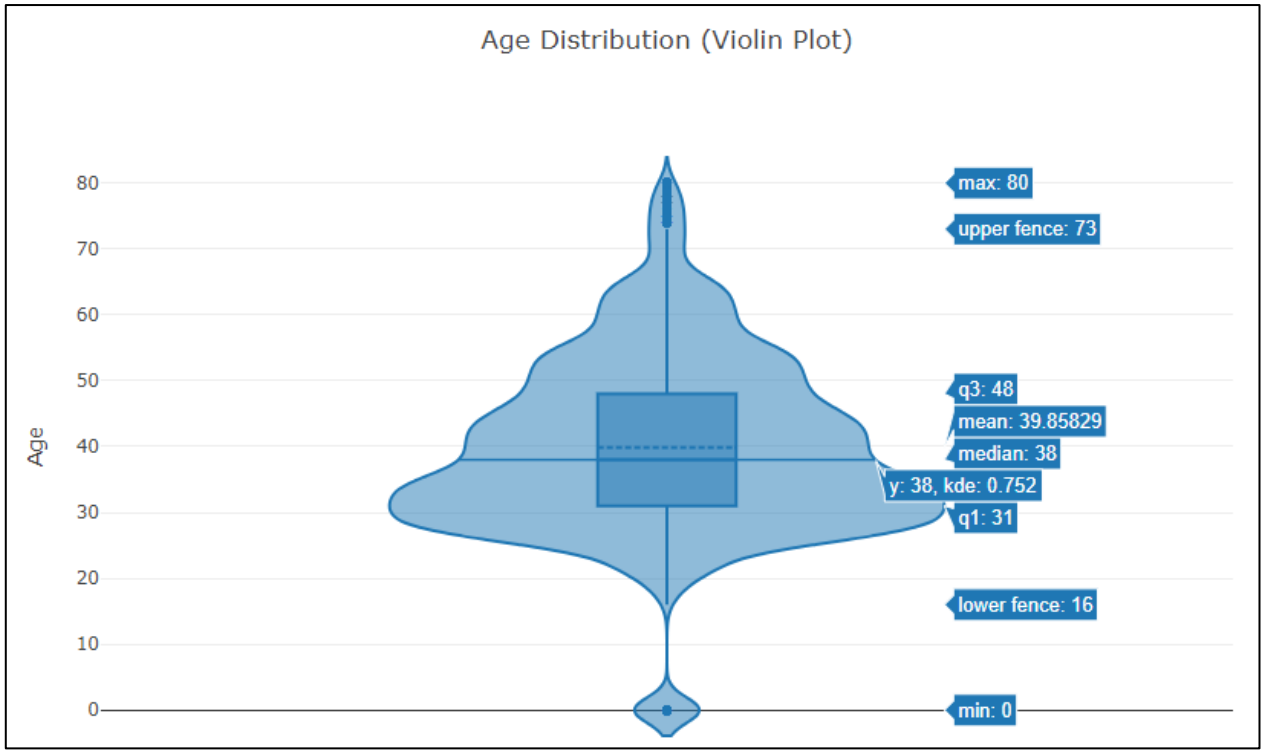
7] Word Cloud

```
d3.csv("../fraud_oracle.csv").then(function(data) {

    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("svg").attr("width", width).attr("height", height);

    const layout = d3.layout.cloud().size([width, height]).words(words)
                    .padding(0.0).rotate(() => ~~(Math.random() * 2) * 90)
                    .fontSize(d => d.size).on("end", draw)
                    .on("word", word => console.log("Word placement:", word));
    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);
    }
});
```

The above word cloud shows the frequency of policy claims as per the vehicle category. Sedan has the highest number of claims and Utility has the lowest number of claims.

8] Violin Plot

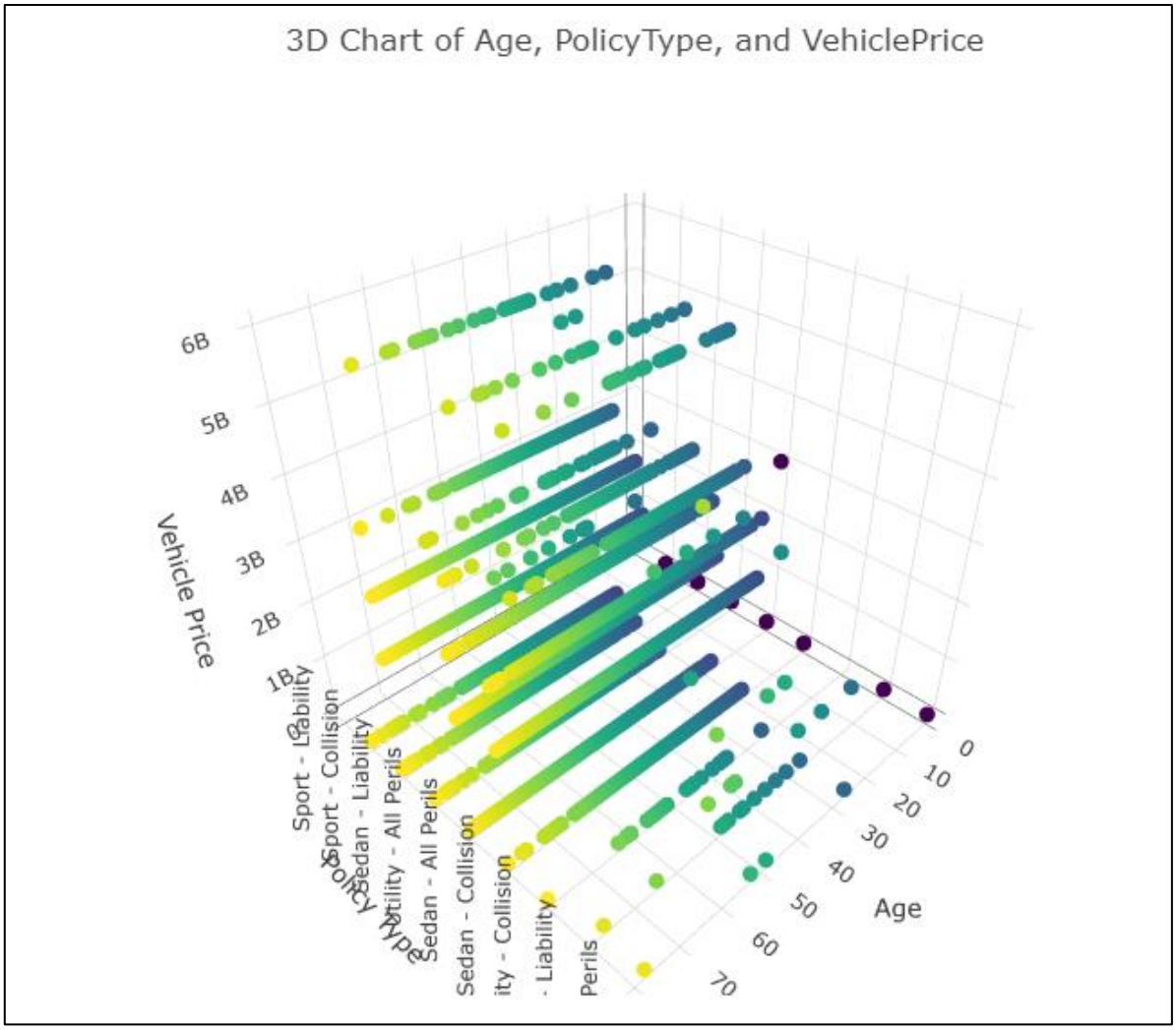
```
Plotly.d3.csv("../fraud_oracle.csv", function(data) {
    const ages = data.map(d => +d.Age);
    const trace = {
        y: ages,
        type: 'violin',
        box: {
            visible: true
        },
        meanline: {
            visible: true
        }
    };

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

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



The above Violin Graph shows distribution as per age. It shows that the maximum number of claims is around the age of 30, 38 being the median, 0 being the lowest and 80 being the highest age of claim.

9] 3D Chart

```
Plotly.d3.csv("../fraud_oracle.csv", function(data) {
    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, PolicyType, and VehiclePrice",
        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);
});
```



The above 3D Graph shows the claims as per age, policy and vehicle price. It can be seen that most cars are in the range from 0 to 3B. There are few entries for age 0 that states it is a part or outlier.

HYPOTHESIS TESTING:

To perform hypothesis testing using the Pearson correlation coefficient to evaluate relationships between numerical variables in the dataset.

```
import pandas as pd
from scipy.stats import pearsonr

df = pd.read_csv('../fraud_oracle.csv')

def convert_vehicle_price(price):
    if "more than" in price:
        return float(price.replace("more than ", ""))
    elif "less than" in price:
        return float(price.replace("less than ", ""))
    elif "to" in price:
        range_values = price.replace(",", "").split(" to ")
        return (float(range_values[0]) + float(range_values[1])) / 2
    else:
        return None

df['VehiclePrice'] = df['VehiclePrice'].apply(convert_vehicle_price)
df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
df = df.dropna(subset=['Age', 'VehiclePrice'])
corr, p_value = pearsonr(df['Age'], df['VehiclePrice'])

print(f"Pearson Correlation between Age and VehiclePrice: {corr}")
print(f"P-value: {p_value}")

alpha = 0.05
if p_value < alpha:
    print(f"Reject the null hypothesis (p-value = {p_value}). There is a significant relationship between Age and VehiclePrice.")
else:
    print(f"Fail to reject the null hypothesis (p-value = {p_value}). There is no significant relationship between Age and VehiclePrice.")
```

OUTPUT:

```
PS C:\Users\SATISH H THAKAR\OneDrive\Desktop\ADV\ADV7> python adv7.py
Pearson Correlation between Age and VehiclePrice: -0.08301855514066347
P-value: 5.3937104478293235e-25
Reject the null hypothesis (p-value = 5.3937104478293235e-25). There is a significant relationship between Age and VehiclePrice.
```

The above relation states that there is a weak negative correlation between Age and Vehicle Price which is then supported using the hypothesis testing as we reject the null hypothesis thus stating that there is a significant relation between Age and Vehicle Price.