NAME:	Divya Suvarna
UID No.	2021200114
BRANCH:	B.E CSE-DS
BATCH:	A
SUBJECT	Advanced Data Visualization
EXPERIMENT No.	7
DATE:	26/10/2024

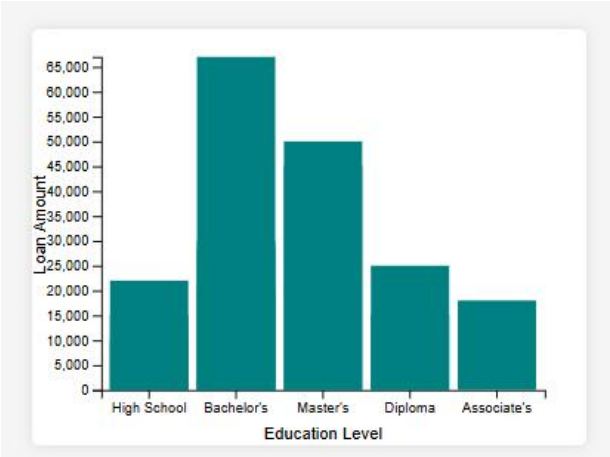
AIM: Experiment Design for Creating Visualizations using D3.js on a Finance Dataset

DATASET: **https://www.kaggle.com/datasets/nikhil1e9/loan-default**

Financial loan services are leveraged by companies across many industries, from big banks to financial institutions to government loans. One of the primary objectives of companies with financial loan services is to decrease payment defaults and ensure that individuals are paying back their loans as expected. In order to do this efficiently and systematically, many companies employ machine learning to predict which individuals are at the highest risk of defaulting on their loans, so that proper interventions can be effectively deployed to the right audience.This dataset has been taken from Coursera's Loan Default Prediction Challenge and will provide you the opportunity to tackle one of the most industry-relevant machine learning problems with a unique dataset that will put your modeling skills to the test. The dataset contains 255,347 rows and 18 columns in total.

ANALYSIS:

1] Bar Graph




```
const svgBar = 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})`);

// Define X and Y axes
const xBar = d3
  .scaleBand()
  .domain(data.map(d => d.education))
  .range([0, width])
  .padding(0.1);
```

```
const yBar = d3
  .scaleLinear()
  .domain([0, d3.max(data, d => d.loanAmount)])
  .range([height, 0]);
```

```
// Append X-axis
svgBar.append("g")
  .attr("transform", `translate(0,${height})`)
  .call(d3.axisBottom(xBar))
  .append("text")
  .attr("x", width / 2)
  .attr("y", 35)
  .attr("fill", "black")
  .attr("text-anchor", "middle")
  .style("font-size", "12px")
  .text("Education Level");
```

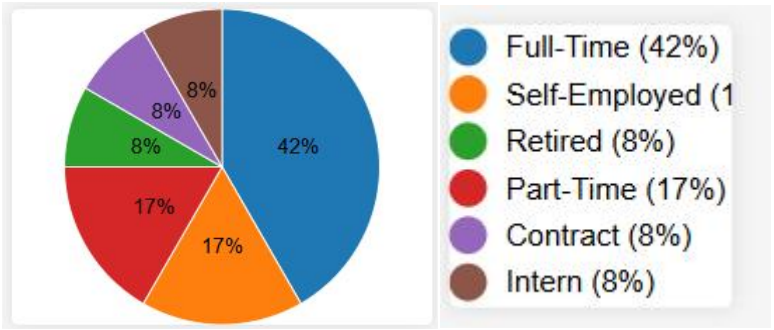
```
// Append Y-axis
svgBar.append("g")
  .call(d3.axisLeft(yBar))
  .append("text")
  .attr("transform", "rotate(-90)")
  .attr("x", -height / 2)
  .attr("y", -40)
  .attr("fill", "black")
  .attr("text-anchor", "middle")
  .style("font-size", "12px")
  .text("Loan Amount");
```

```
// Create the tooltip
const tooltip = d3.select("body")
  .append("div")
  .style("position", "absolute")
  .style("background-color", "white")
  .style("border", "1px solid #d3d3d3")
  .style("border-radius", "5px")
  .style("padding", "8px")
  .style("display", "none") // Hide tooltip initially
  .style("pointer-events", "none") // Avoid conflicts with hover
  .style("box-shadow", "0px 0px 10px rgba(0, 0, 0, 0.2)"); // Add shadow for better visibility
```

```
// Append bars to the chart
svgBar.selectAll(".bar")
  .data(data)
  .enter()
  .append("rect")
  .attr("class", "bar")
  .attr("x", d => xBar(d.education))
  .attr("y", d => yBar(d.loanAmount))
  .attr("width", xBar.bandwidth())
  .attr("height", d => height - yBar(d.loanAmount))
  .attr("fill", "teal")
  .on("mouseover", function (event, d) {
    tooltip
      .style("display", "block")
      .html(`Loan Amount: $${d.loanAmount}`);
    d3.select(this).attr("fill", "darkblue"); // Change color on hover
  })
  .on("mousemove", function (event) {
    tooltip
      .style("left", (event.pageX + 10) + "px")
      .style("top", (event.pageY - 30) + "px"); // Adjust top positioning
  })
  .on("mouseout", function () {
    tooltip.style("display", "none");
    d3.select(this).attr("fill", "teal"); // Reset color on mouseout
  });
```

The above Bar Graph shows the loan amounts by education level, with bachelor's degree holders having the highest average loan amount.

2] Pie Graph




```
// 2. Pie Chart - Default Status
const svgPie = d3
  .select("#pie-chart")
  .append("svg")
  .attr("width", width)
  .attr("height", height)
  .append("g")
  .attr("transform", `translate(${width / 2},${height / 2})`);

// Count occurrences of each employment type
const employmentData = d3.rollup(
  data,
  v => v.length,
  d => d.employmentType
);
```

```
const pie = d3.pie().value(d => d[1]); // d[1] contains the count
const arc = d3.arc().innerRadius(0).outerRadius(Math.min(width, height) / 2);
```

```
// Create arcs and add them to the pie chart
svgPie
  .selectAll("path")
  .data(pie(Array.from(employmentData.entries())))
  .enter()
  .append("path")
  .attr("d", arc)
  .attr("fill", (d, i) => d3.schemeCategory10[i]) // Use a color scheme for diversity
  .attr("stroke", "white")
  .attr("stroke-width", 1);
```

```
// Add percentage labels
svgPie
  .selectAll("text")
  .data(pie(Array.from(employmentData.entries())))
  .enter()
  .append("text")
  .attr("transform", d => `translate(${arc.centroid(d)})`)
  .attr("dy", "0.35em")
  .attr("text-anchor", "middle")
  .text(d => `${Math.round((d.data[1] / d3.sum(Array.from(employmentData.values()))) * 100)}%`);
```

```
// Add legend
const legend = d3.select("#pie-chart")
  .append("svg")
  .attr("width", 150) // Adjust as necessary
  .attr("height", 150) // Adjust as necessary
  .attr("transform", `translate(${width - 170}, 20)`); // Position the legend
```

```
// Create legend items
const legendItems = legend
  .selectAll(".legend")
  .data(Array.from(employmentData.entries()))
  .enter()
  .append("g")
  .attr("class", "legend")
  .attr("transform", (d, i) => `translate(0, ${i * 25})`); // Space out legend items
```

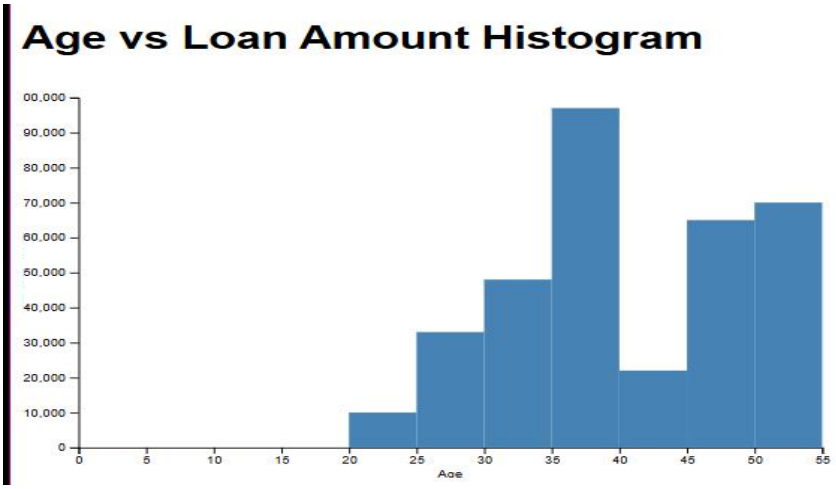
```
legendItems
  .append("circle")
  .attr("cx", 10) // X position of the circle
  .attr("cy", 12) // Y position of the circle
  .attr("r", 10) // Radius of the circle
  .style("fill", (d, i) => d3.schemeCategory10[i]); // Use the same color as the pie slices
```

```
legendItems
```

```
  .append("text")
  .attr("x", 30) // X position of the text
  .attr("y", 12) // Align vertically with the circle
  .attr("dy", "0.35em") // Adjust vertical alignment
  .text(d => `${d[0]} (${Math.round((d[1] / d3.sum(Array.from(employmentData.values()))) * 100)}%)`); // Employment Type and Percentage
```

The above pie chart shows the relationship between income and loan amount, with green and red points possibly indicating different categories or groups.

3] Histogram




```
const svg = d3.select("#histogram")
            .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})`);

        // Create histogram bins based on age
        const histogram = d3.histogram()
            .value(d => d.age) // The value accessor for the histogram
            .domain([0, d3.max(data, d => d.age)]) // Set the domain
            .thresholds(10); // Number of bins
```

```
        const bins = histogram(data);
```

```
        // Set x scale for age using linear scale
        const x = d3.scaleLinear()
            .domain([0, d3.max(data, d => d.age)]) // Domain is the range of ages
            .range([0, width]);
```

```
        // Set y scale for loan amount
        const y = d3.scaleLinear()
            .domain([0, d3.max(bins, d => d3.sum(d, bin => bin.loanAmount))]) // Sum of loan amounts in each bin
            .nice() // Nice round the domain
            .range([height, 0]);
```

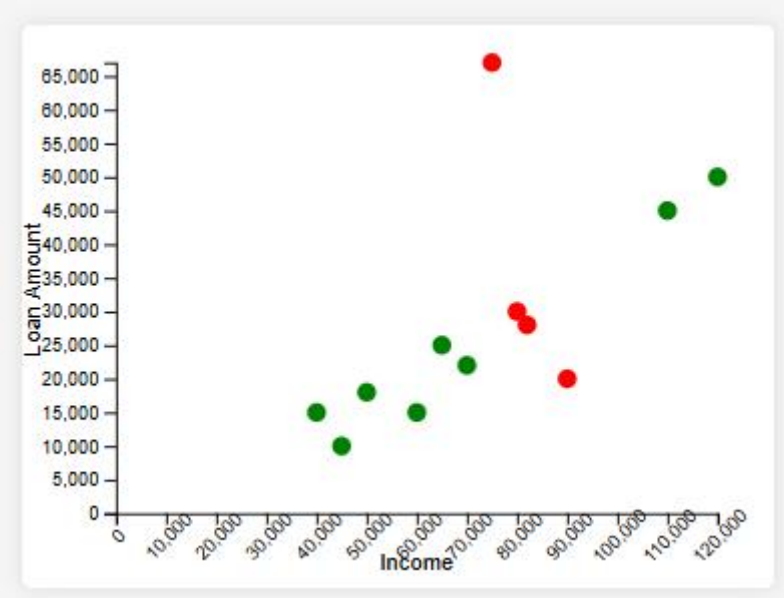
```
        // Append x axis
        svg.append("g")
            .attr("transform", `translate(0,${height})`)
            .call(d3.axisBottom(x).ticks(10)) // Adjust the number of ticks as necessary
            .append("text")
            .attr("x", width / 2)
            .attr("y", margin.bottom)
            .attr("fill", "black")
            .attr("text-anchor", "middle")
            .text("Age");
```

```
        // Append y axis
        svg.append("g")
            .call(d3.axisLeft(y))
            .append("text")
            .attr("transform", "rotate(-90)")
            .attr("x", -height / 2)
            .attr("y", -margin.left)
            .attr("fill", "black")
            .attr("text-anchor", "middle")
            .text("Total Loan Amount");
```

```
        // Create bars
        svg.selectAll(".bar")
            .data(bins)
            .enter()
            .append("rect")
            .attr("class", "bar")
            .attr("x", d => x(d.x0)) // Start of each bin
            .attr("y", d => y(d3.sum(d, bin => bin.loanAmount))) // Use total loan amount for the height
            .attr("width", d => x(d.x1) - x(d.x0)) // Width is the difference between bin ends
            .attr("height", d => height - y(d3.sum(d, bin => bin.loanAmount)));
```

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] Scatter plot




```
const svgScatter = 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 xScatter = d3.scaleLinear().domain([0, d3.max(data, d => d.income)]).range([0, width]);
const yScatter = d3.scaleLinear().domain([0, d3.max(data, d => d.loanAmount)]).range([height, 0]);
```

```
// Add X axis with labels
const xAxis = svgScatter.append("g").attr("transform", `translate(0,${height})`).call(d3.axisBottom(xScatter));
```

```
// Add X axis label
svgScatter.append("text")
  .attr("class", "x-axis-label")
  .attr("x", width / 2) // Center the label horizontally
  .attr("y", height + margin.bottom - 10) // Position below the x-axis
  .attr("fill", "black")
  .attr("text-anchor", "middle")
  .style("font-size", "12px")
  .text("Income");
```

```
// Rotate x-axis labels
xAxis.selectAll("text")
  .attr("transform", "rotate(-45)") // Rotate the labels by -45 degrees
  .attr("dy", "0.35em") // Adjust vertical alignment
  .attr("dx", "-0.8em"); // Adjust horizontal position to avoid overlap
```

```
// Add Y axis with labels
const yAxis = svgScatter.append("g").call(d3.axisLeft(yScatter));
```

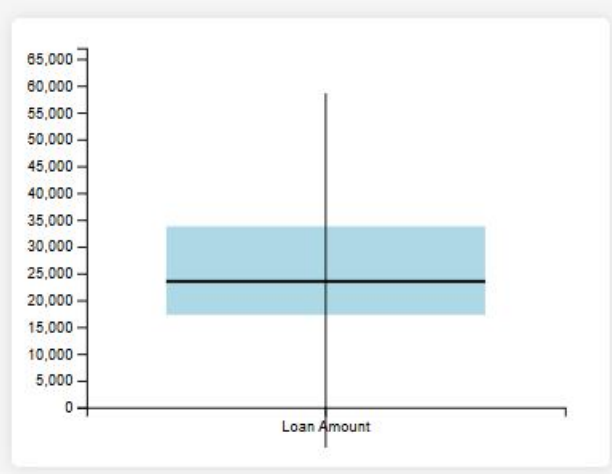
```
// Add Y axis label
```

```
svgScatter.append("text")
  .attr("class", "y-axis-label")
  .attr("transform", "rotate(-90)") // Rotate the label for vertical alignment
  .attr("y", -margin.left + 10) // Position to the left of the Y-axis
  .attr("x", -height / 2) // Center it vertically
  .attr("fill", "black")
  .attr("text-anchor", "middle")
  .style("font-size", "12px")
  .text("Loan Amount");
```

```
// Append circles for scatter plot
svgScatter.selectAll("circle")
  .data(data)
  .enter()
  .append("circle")
  .attr("cx", d => xScatter(d.income))
  .attr("cy", d => yScatter(d.loanAmount))
  .attr("r", 5) // Set radius for the circles
  .attr("fill", d => (d.default ? "red" : "green")) // Color based on default status
  .on("mouseover", function (event, d) {
    tooltip
      .style("display", "block")
      .html(`Income: $${d.income}<br>Loan Amount: $${d.loanAmount}<br>Default: ${d.default ? "Yes" : "No"}`);
    d3.select(this).attr("fill", "orange"); // Change color on hover
  })
  .on("mousemove", function (event) {
    tooltip
      .style("left", (event.pageX + 10) + "px")
      .style("top", (event.pageY - 30) + "px");
  })
  .on("mouseout", function () {
    tooltip.style("display", "none");
    d3.select(this).attr("fill", d => (d.default ? "red" : "green")); // Reset color on mouseout
  });
```

This histogram shows the distribution of loan amounts across different age groups, with the highest loan amounts concentrated around ages 35 to 45.

5] Box Plot




```
const svgBox = d3
  .select("#box-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})`);

// Calculate the quartiles and the interquartile range
const q1 = d3.quantile(data.map(d => d.loanAmount).sort(d3.ascending), 0.25);
const median = d3.median(data.map(d => d.loanAmount));
const q3 = d3.quantile(data.map(d => d.loanAmount).sort(d3.ascending), 0.75);
const iqr = q3 - q1;
```

```
// Define the scales for the box plot
const xBox = d3.scaleBand().domain(["Loan Amount"]).range([0, width]).padding(0.2);
const yBox = d3.scaleLinear().domain([0, d3.max(data, d => d.loanAmount)]).range([height, 0]);
```

```
// Append the rectangle for the box
```

```
svgBox.append("rect")
  .attr("x", xBox("Loan Amount"))
  .attr("y", yBox(q3))
  .attr("width", xBox.bandwidth())
  .attr("height", yBox(q1) - yBox(q3))
  .attr("fill", "lightblue");
```

```
// Add median line
svgBox.append("line")
  .attr("x1", xBox("Loan Amount"))
  .attr("x2", xBox("Loan Amount") + xBox.bandwidth())
  .attr("y1", yBox(median))
  .attr("y2", yBox(median))
  .attr("stroke", "black")
  .attr("stroke-width", 2);
```

```
// Append whiskers
svgBox.append("line")
  .attr("x1", xBox("Loan Amount") + xBox.bandwidth() / 2)
  .attr("x2", xBox("Loan Amount") + xBox.bandwidth() / 2)
  .attr("y1", yBox(q1 - 1.5 * iqr))
  .attr("y2", yBox(q3 + 1.5 * iqr))
  .attr("stroke", "black");
```

```
// Y-axis for box plot
svgBox.append("g")
  .call(d3.axisLeft(yBox));
```

```
// X-axis for box plot
svgBox.append("g")
  .attr("transform", `translate(0,${height})`)
  .call(d3.axisBottom(xBox));
```

The box plot shows the range and median of loan amounts, highlighting potential outliers and the spread of loan values.

6]Word Cloud


```
// Create word cloud layout
        const layout = d3.layout.cloud()
            .size([width, height])
            .words(employmentFrequency.map(([word, size]) => ({ text: word, size: size * 20 })))
            .padding(5)
            .rotate(() => (Math.random() > 0.5 ? 0 : 90)) // Random rotation for visual variety
            .fontSize(d => d.size)
            .on("end", draw);

        layout.start();
```

```
        // Draw word cloud
        function draw(words) {
            d3.select("#wordcloud")
                .append("svg")
                .attr("width", width)
                .attr("height", height)
                .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);
    }
```

This word cloud illustrates the prevalence of different employment types, with "Full-Time" being the most prominent, indicating it's the most common employment type among the group.

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 Loan Amount 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 Loan Amount.