

Husain Mehdi

F20DV

Lab 2

Demonstrated 25 Feb 2022 to Amit Parekh

EX1

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5
6      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
7      <style>
8          div {
9              display: inline-block;
10             width: 10px;
11             height: 10px;
12             margin: 1px;
13             padding: 10px;
14         }
15     </style>
16     <script>
17
18         var body = d3.select('body');
19         var myColor1 = d3.scaleSequential().domain([1, 10]).interpolator(d3.interpolateCool);
20
21         // Set Dimensions
22         const xSize = 600; const ySize = 600;
23         const margin = 40;
24         const xMax = xSize - margin * 2;
25         const yMax = ySize - margin * 2;
26
27         // Create Random Points
28         const numPoints = 100;
29         const data = [];
30         for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.sin(6.2 * i / 100) }); }
31
32
33
34
35         // Get the 'limits' of the data - the full extent (mins and max)
36         // so the plotted data fits perfectly
37         const xExtent = d3.extent(data, d => { return d.x });
38         const yExtent = d3.extent(data, d => { return d.y });
39
40         // Append SVG Object to the Page
41         const svg = d3.select("body")
42             .append("svg")
43             .attr('width', xSize)
44             .attr('height', ySize)
45             .append("g")
46             .attr("transform", "translate(" + margin + "," + margin + ")");
47
48         // X Axis
49         const x = d3.scaleLinear()
50             .domain([xExtent[0], xExtent[1]])
51             .range([0, xMax]);
52
53         // bottom
54         svg.append("g")
55             .attr("transform", "translate(0," + yMax + ")")
56             .call(d3.axisBottom(x))
```

```
56          // top
57          svg.append("g")
58          | .call(d3.axisTop(x));
59
60          // Y Axis
61          const y = d3.scaleLinear()
62          | .domain([yExtent[0], yExtent[1]])
63          | .range([yMax, 0]);
64
65          // left y axis
66          svg.append("g")
67          | .call(d3.axisLeft(y));
68
69          // right y axis
70          svg.append("g")
71          | .attr("transform", `translate(${yMax},0`)
72          | .call(d3.axisRight(y));
73
74          // Add the line
75          svg.append("path")
76          | .datum(data)
77          | .attr("fill", "none")
78          | .attr("stroke", "steelblue")
79          | .attr("stroke-width", 1.5)
80          | .attr("d", d3.line()
81          | | .x(function (d) { return x(d.x) })
82          | | .y(function (d) { return y(d.y) })
83          );
84
85          var triangle = d3.symbol()
86          | .type(d3.symbolTriangle)
87          | .size(50)
88
89          svg.selectAll("dot")
90          | .data(data)
91          | .enter()
92          | .append("circle")
93          | .attr("cx", function (d) { return x(d.x) })
94          | .attr("cy", function (d) { return y(d.y) })
95          | .attr("r", 5)
```

```
110 |         transform-origin: 0% 0%;  
111 |         position: relative;  
112 |     }  
113 |     .pulse:hover  
114 |     {  
115 |         -webkit-animation-name: pulsar;  
116 |         -webkit-animation-duration: 1s;  
117 |         -webkit-animation-iteration-count: infinite;  
118 |         -webkit-animation-direction: alternate;  
119 |         animation-name: pulsar;  
120 |         animation-duration: 1s;  
121 |         animation-iteration-count: infinite;  
122 |         animation-direction: alternate;  
123 |         -webkit-transform-origin: 0% 0%;  
124 |         transform-origin: 0% 0%;  
125 |     }  
126 |     @keyframes pulsar {  
127 |         from {  
128 |             }  
129 |             to {  
130 |                 }  
131 |                 r: 5%;  
132 |                 transform-origin: 0% 0%;  
133 |                 z-index: -100;  
134 |             }  
135 |         }  
136 |     </style>  
137 |     <svg class='pulse' width='30' height='30'></svg>  
138 |  
139 | </body>  
140 |
```

EX2

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <style>
6          .b {
7              padding: 20px;
8          }
9
10         .c {
11             background-color: yellow;
12             padding: 20px;
13             display: none;
14         }
15
16
17         .b:hover .c {
18             position: absolute;
19             display: inline-block;
20         }
21     </style>
22     <script src='https://d3js.org/d3.v7.min.js'></script>
23
24
25     <script>
26         var myData = ['a', 'b', 'c', 'd'];
27
28         d3.select("body")
29             .selectAll("div")
30             .data(myData)
31             .enter()
32             .append('div')
33             .attr("class", "b")
34             .text(function (d, i) {
35                 return d;
36             })
37             .append('div')
38             .attr("class", "c")
39             .text("hovertext")
40
41
```

EX3

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6
7  <script>
8      d3.select('body')
9          .append('div')
10         .text("lorum ipsum")
11         .style('width', '100px')
12         .style('height', '20px')
13         .style('background-color', 'green')
14         .style("color", "black");
15
16     d3.selectAll("div")
17         .on("mouseover", function (event) {
18             d3.select(this)
19                 .style("background-color", "orange")
20                 .style("height", "100px")
21                 .style("color", "red")
22                 .style("border-style", "double")
23                 .style("border-width", "5px");
24
25             // Get current event info
26             // Note: d3.event (event) passed as the first argument to all listeners
27             console.log(event);
28
29             // Get x & y co-ordinates
30             // Note: d3.mouse was removed in d3v6, you should use d3.pointer(event)
31             console.log(d3.pointer(event));
32         })
33         .on("mouseout", function () {
34             d3.select(this)
35                 .style("background-color", "steelblue")
36                 .style("height", "20px")
37                 .style("color", "blue")
38                 .style("border-style", "none");
39         });
40     </script>
41 </body>
```

EX4

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6
7      <script>
8          var svg = d3.select("body")
9              .append("svg")
10             .attr("width", 400)
11             .attr("height", 400)
12             .style("border", '1px solid green');
13
14          svg.append("circle")
15             .attr("cx", 200)
16             .attr("cy", 200)
17             .attr("r", 10)
18             .attr("fill", "green");
19
20          d3.selectAll("circle")
21             .on("mouseover", function (event) {
22                 d3.select(this)
23                     .attr("r", 100)
24                     .attr("fill", "red");
25
26                 // Get current event info
27                 // Note: d3.event (event) passed as the first argument to all listeners
28                 console.log(event);
29
30                 // Get x & y co-ordinates
31                 // Note: d3.mouse was removed in d3v6, you should use d3.pointer(event)
32                 console.log(d3.pointer(event));
33             })
34             .on("mouseout", function () {
35                 d3.select(this)
36                     .attr("r", 10)
37                     .attr("fill", "green")
38             });
39
40      </script>
41  </body>
```

EX5

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6
7      <script>
8
9          var div = d3.select("body").append("div")
10         .attr("class", "tooltip")
11         .style("opacity", 0);
12
13         var svg = d3.select("body")
14         .append("svg")
15         .attr("width", 400)
16         .attr("height", 400)
17         .style("border", '1px solid green')
18         .on('mousemove', (event) => {
19             var coords = d3.pointer(event);
20             console.log(coords[0], coords[1]) // log the mouse x,y position
21             div.transition()
22                 .duration(200)
23                 .style("opacity", .9);
24             div.html(coords[0] + ", " + coords[1])
25                 .style("left", (coords[0]) + "px")
26                 .style("top", (coords[1]) + "px");
27         });
28
29         svg.append("circle")
30         .attr("cx", 200)
31         .attr("cy", 200)
32         .attr("r", 10)
33         .attr("fill", "green")
```

```
34
35
36     d3.selectAll("circle")
37         .on("mouseover", function (event) {
38             d3.select(this)
39                 .attr("r", 100)
40                 .attr("fill", "red")
41             e => console.log(d3.pointer(e));
42
43             // Get current event info
44             // Note: d3.event (event) passed as the first argument to all listeners
45             console.log(event);
46
47             // Get x & y co-ordinates
48             // Note: d3.mouse was removed in d3v6, you should use d3.pointer(event)
49             console.log(d3.pointer(event));
50         })
51         .on("mouseout", function () {
52             d3.select(this)
53                 .attr("r", 10)
54                 .attr("fill", "green");
55         });
56     </script>
57     <style>
58         div.tooltip {
59             position: absolute;
60             text-align: center;
61             width: 60px;
62             height: 28px;
63             padding: 2px;
64             font: 12px sans-serif;
65             background: lightsteelblue;
66             border: 0px;
67             border-radius: 8px;
68             pointer-events: none;
69         }
70     </style>
71 </body>
```

EX6

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7          d3.select('body')
8              .append("div")
9              .style('width', '100px')
10             .style('height', '100px')
11             .style('background-color', 'blue')
12             .transition()
13             .duration(1000)
14             .style("background-color", "red")
15             .transition()
16             .duration(2000)
17             .style("background-color", "green");
18
19      </script>
20  </body>
```

EX7

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7          d3.select('body')
8              .append("div")
9              .style('width', '100px')
10             .style('height', '100px')
11             .style('background-color', 'blue')
12             .transition()
13             .duration(1000)
14             .style("background-color", "red")
15             .any('width', '50px')
16             .style('height', '50px')
17             .transition()
18             .duration(2000)
19             .style("background-color", "green")
20             .style('width', '200px')
21             .style('height', '200px');
22
23      </script>
24  </body>
```

EX8

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7          d3.select('body')
8              .append("div")
9                  .style('width', '100px')
10                 .style('height', '100px')
11                 .style('background-color', 'blue');
12
13
14
15         d3.selectAll("div")
16             .on("mouseover", function (event) {
17                 d3.select(this).transition()
18                     .duration(1000)
19                     .style("background-color", "red")
20                     .style('width', '50px')
21                     .style('height', '50px');
22             })
23             .on("mouseout", function () {
24                 d3.select(this).transition()
25                     .duration(2000)
26                     .style("background-color", "green")
27                     .style('width', '200px')
28                     .style('height', '200px')})
29
30     </script>
31 </body>
```

EX9

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7          d3.select('body')
8              .append("div")
9              .style('width', '100px')
10             .style('height', '100px')
11             .style('background-color', 'blue')
12             .style('transform', 'scale(1.0)')
13             .transition()
14             .ease(d3.easeBounce)
15             .duration(1000)
16             .style("background-color", "red")
17             .style('transform', 'scale(0.5)')
18
19
20          d3.select('body')
21              .append("div")
22              .style('width', '100px')
23              .style('height', '100px')
24              .style('background-color', 'blue')
25              .style('transform', 'scale(1.0)')
26              .transition()
27              .ease(d3.easeQuadOut)
28              .duration(1000)
29              .style("background-color", "red")
30              .style('transform', 'scale(0.5)')
31
32
33          d3.select('body')
34              .append("div")
35              .style('width', '100px')
36              .style('height', '100px')
37              .style('background-color', 'blue')
38              .style('transform', 'scale(1.0)')
39              .transition()
40              .ease(d3.easeElasticIn.amplitude(1).period(0.3))
41              .duration(1000)
42              .style("background-color", "red")
43              .style('transform', 'scale(0.5)')
44
45      </script>
46
47  </body>
```

EX10

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7
8          var svg = d3.select("body")
9              .append("svg")
10             .attr("width", 400)
11             .attr("height", 400)
12             .style("border", '1px solid green');
13
14          svg.append("circle")
15             .attr("cx", 200)
16             .attr("cy", 200)
17             .attr("r", 10)
18             .attr("fill", "green");
19
20          d3.selectAll("circle")
21              .on("mouseover", function (event) {
22                  d3.select(this)
23                      .transition()
24                      .ease(d3.easeBounce)
25                      .duration(1000)
26                      .attr("r", 100)
27                      .attr("fill", "red");
28              })
29              .on("mouseout", function () {
30                  d3.select(this)
31                      .transition()
32                      .ease(d3.easeBounce)
33                      .duration(1000)
34                      .attr("r", 10)
35                      .attr("fill", "green")
36              });
37
38      </script>
39  </body>
```

EX11

```
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7          var svg = d3.select("body")
8              .append("svg")
9                  .attr("width", 400)
10                 .attr("height", 400)
11                 .style("border", '1px solid green');
12
13         svg.append("circle")
14             .attr("cx", 200)
15             .attr("cy", 200)
16             .attr("r", 10)
17             .attr("fill", "green");
18
19         svg.append("text")
20             .attr("x", 200)
21             .attr("y", 350)
22             .text("hi");
23
24         d3.selectAll("circle")
25             .on("mouseover", function (event) {
26                 d3.select(this)
27                     .transition()
28                     .ease(d3.easeBounce)
29                     .duration(1000)
30                     .attr("r", 100)
31                     .attr("fill", "red");
32             })
33             .on("mouseout", function () {
34                 d3.select(this)
35                     .transition()
36                     .ease(d3.easeBounce)
37                     .duration(1000)
38                     .attr("r", 10)
39                     .attr("fill", "green")
40             });
41
42         d3.selectAll("text")
43             .on("mouseover", function (event) {
44                 d3.select(this)
45                     .transition()
46                     .ease(d3.easeBounce)
47                     .duration(1000)
48                     .attr("font-size", 100)
49                     .attr("fill", "red");
50             })
51             .on("mouseout", function () {
52                 d3.select(this)
53                     .transition()
54                     .ease(d3.easeBounce)
55                     .duration(1000)
56                     .attr("font-size", 10)
57                     .attr("fill", "green")
58             });

```

EX12

```
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7
8          var svg = d3.select("body")
9              .append("svg")
10             .attr("width", 500)
11             .attr("height", 500);
12
13         var bar1 = svg.append("rect")
14             .attr("fill", "blue")
15             .attr("x", 100)
16             .attr("y", 20)
17             .attr("height", 20)
18             .attr("width", 10)
19
20         var bar2 = svg.append("rect")
21             .attr("fill", "blue")
22             .attr("x", 120)
23             .attr("y", 20)
24             .attr("height", 20)
25             .attr("width", 10)
26
27         var bar3 = svg.append("rect")
28             .attr("fill", "blue")
29             .attr("x", 140)
30             .attr("y", 20)
31             .attr("height", 20)
32             .attr("width", 10)
33
34
35         update();
36
37         function update() {
38             bar1.transition()
39                 .ease(d3.easeLinear)
40                 .duration(2000)
41                 .attr("height", 100)
42
43             bar2.transition()
44                 .ease(d3.easeLinear)
45                 .duration(2000)
46                 .delay(2000)
47                 .attr("height", 100)
48
49
50             bar3.transition()
51                 .ease(d3.easeLinear)
52                 .duration(2000)
53                 .delay(4000)
54                 .attr("height", 100)
55         }
56     </script>
57 </body>
```

Ex13

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7
8          var svg = d3.select("body")
9              .append("svg")
10             .attr("width", 500)
11             .attr("height", 500);
12
13         var bar1 = svg.append("rect")
14             .attr("fill", "blue")
15             .attr("x", 100)
16             .attr("y", 20)
17             .attr("height", 20)
18             .attr("width", 10)
19
20         var bar2 = svg.append("rect")
21             .attr("fill", "blue")
22             .attr("x", 120)
23             .attr("y", 20)
24             .attr("height", 20)
25             .attr("width", 10)
26
27         var bar3 = svg.append("rect")
28             .attr("fill", "blue")
29             .attr("x", 140)
30             .attr("y", 20)
31             .attr("height", 20)
32             .attr("width", 10)
33
34
35     update();
36
37     function update() {
38         bar1.transition()
39             .ease(d3.easeLinear)
40             .duration(2000)
41             .attr("height", 100)
42             .end()
43             .then(function() {
44                 bar1.transition()
45                     .ease(d3.easeLinear)
46                     .duration(2000)
47                     .attr("height", 20);})
48
49         bar2.transition()
50             .ease(d3.easeLinear)
51             .duration(2000)
52             .delay(2000)
53             .attr("height", 100)
54             .end()
55             .then(function() {
```

```
57          .ease(d3.easeLinear)
58          .duration(2000)
59          .attr("height", 20);})
60
61      bar3.transition()
62          .ease(d3.easeLinear)
63          .duration(2000)
64          .delay(4000)
65          .attr("height", 100)
66          .end()
67          .then(function() {
68              bar3.transition()
69                  .ease(d3.easeLinear)
70                  .duration(2000)
71                  .attr("height", 20);})
72      }
73
74  </script>
75  </body>
```

EX14

```
1   <!DOCTYPE html>
2   <html lang="en">
3
4   <body>
5     <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6   </script>
7
8     var svg = d3.select("body")
9       .append("svg")
10      .attr("width", 500)
11      .attr("height", 500);
12
13    var bar1 = svg.append("rect")
14      .attr("fill", "blue")
15      .attr("x", 100)
16      .attr("y", 20)
17      .attr("height", 20)
18      .attr("width", 10)
19
20    var bar2 = svg.append("rect")
21      .attr("fill", "blue")
22      .attr("x", 120)
23      .attr("y", 20)
24      .attr("height", 20)
25      .attr("width", 10)
26
27    var bar3 = svg.append("rect")
28      .attr("fill", "blue")
29      .attr("x", 140)
30      .attr("y", 20)
31      .attr("height", 20)
32      .attr("width", 10)
33
34
35     update();
36
37   function update() {
38     bar1.transition()
39       .ease(d3.easeLinear)
40       .duration(2000)
41       .attr("height", 100)
42       .attr("fill", "red")
43       .end()
44       .then(function () {
45         bar1.transition()
46           .ease(d3.easeLinear)
47           .duration(2000)
48           .attr("height", 20)
49           .attr("fill", "blue")
50           ;
51       })
52
53     bar2.transition()
54       .ease(d3.easeLinear)
55       .duration(2000)
56       .attr("height", 100)
57       .attr("fill", "red")
58       .end()
59       .then(function () {
60         bar2.transition()
61           .ease(d3.easeLinear)
62           .duration(2000)
63           .attr("height", 20)
64           .attr("fill", "blue")
65           ;
66       })
67
68     bar3.transition()
69       .ease(d3.easeLinear)
70       .duration(2000)
71       .attr("height", 100)
72       .attr("fill", "red")
73       .end()
74       .then(function () {
75         bar3.transition()
76           .ease(d3.easeLinear)
77           .duration(2000)
78           .attr("height", 20)
79           .attr("fill", "blue")
80           ;
81       })
82
83     update();
84   }
85
86
87   update();
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
55         .duration(2000)
56         .delay(2000)
57         .attr("height", 100)
58         .attr("fill", "red")
59         .end()
60         .then(function () {
61             bar2.transition()
62                 .ease(d3.easeLinear)
63                 .duration(2000)
64                 .attr("height", 20)
65                 .attr("fill", "blue")
66                 ;
67         })
68
69     bar3.transition()
70         .ease(d3.easeLinear)
71         .duration(2000)
72         .delay(4000)
73         .attr("height", 100)
74         .attr("fill", "red")
75         .end()
76         .then(function () {
77             bar3.transition()
78                 .ease(d3.easeLinear)
79                 .duration(2000)
80                 .attr("height", 20)
81                 .attr("fill", "blue")
82                 ;
83         })
84     }
85
86     </script>
87 </body>
```

EX15

```
4  <body>
5
6      <style>
7          .bar {
8              fill: steelblue;
9          }
10
11         .highlight {
12             fill: orange;
13         }
14     </style>
15
16     <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
17
18     <svg width="600" height="500"></svg>
19
20     <script>
21         var svg = d3.select("svg");
22         var margin = 200;
23         var width = svg.attr("width") - margin;
24         var height = svg.attr("height") - margin;
25
26         svg.append("text")
27             .attr("transform", "translate(100,0)")
28             .attr("x", 50)
29             .attr("y", 50)
30             .attr("font-size", "24px")
31             .text("Stock Price")
32
33         var x = d3.scaleBand().range([0, width]).padding(0.4);
34         var y = d3.scaleLinear().range([height, 0]);
35         var g = svg.append("g")
36             .attr("transform", "translate(" + 100 + "," + 100 + ")");
37
38         d3.csv("https://raw.githubusercontent.com/HusainMehdi/DataVis/main/csvfile.csv").then(function (data) {
39             x.domain(data.map(function (d) { return d.year; }));
40             y.domain([0, d3.max(data, function (d) { return d.value; })]);
41             g.append("g")
42                 .attr("transform", "translate(0," + height + ")")
43                 .call(d3.axisBottom(x))
44                 .append("text")
45                 .attr("y", height - 250)
46                 .attr("x", width - 100)
47                 .attr("text-anchor", "end")
48                 .attr("stroke", "black")
49                 .text("Year");
50             g.append("g")
51                 .call(d3.axisLeft(y).tickFormat(function (d) {
52                     return "$" + d;
53                 }).ticks(10))
54                 .append("text")
55                 .attr("transform", "rotate(-90)")
56                 .attr("y", 6)
57                 .attr("dy", "-5.1em")
58                 .attr("text-anchor", "end")
```

```

59         .attr("stroke", "black")
60         .text("Stock Price");
61     g.selectAll(".bar")
62     .data(data)
63     .enter().append("rect")
64     .attr("class", "bar")
65     // .on(... ) - call mouse events here...
66
67     .on("mouseover", function onMouseOver(d, i) {
68         d3.select(this).attr('class', 'highlight');
69         d3.select(this)
70             .transition() // adds animation
71             .duration(400)
72             .attr('width', x.bandwidth() + 5)
73             .attr("y", function (d) { return y(d.value) - 10; })
74             .attr("height", function (d) { return height - y(d.value) + 10; });
75         g.append("text")
76             .attr('class', 'val')
77             .attr('x', function () {
78                 return x(d.year);
79             })
80             .attr('y', function () {
81                 return y(d.value) - 15;
82             })
83             .text(function (d) { return '$' + i.value; }); // Value of the text
84     })
85
86     .on("mouseout", function onMouseOut(d, i) {
87         // use the text label class to remove label onmouseout
88         d3.select(this).attr('class', 'bar');
89         d3.select(this)
90             .transition() // adds animation
91             .duration(400)
92             .attr('width', x.bandwidth())
93             .attr("y", function (d) { return y(i.value); })
94             .attr("height", function (d) { return height - y(i.value); });
95         d3.selectAll('.val')
96             .remove()
97     })
98
99
100    .attr("x", function (d) { return x(d.year); })
101    .attr("y", function (d) { return y(d.value); })
102    .attr("width", x.bandwidth())
103    .transition()
104    .ease(d3.easeLinear)
105    .duration(400)
106    .delay(function (d, i) {
107        return i * 50;
108    })
109    .attr("height", function (d) { return height - y(d.value); });
110
111
112
113     </script>
114 </body>

```

EX16

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5
6    <style>
7      .bar {
8        fill: steelblue;
9      }
10
11     .highlight {
12       fill: orange;
13     }
14   </style>
15
16   <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
17
18   <svg width="600" height="500"></svg>
19
20  <script>
21    var svg = d3.select("svg");
22    var margin = 200;
23    var width = svg.attr("width") - margin;
24    var height = svg.attr("height") - margin;
25
26    svg.append("text")
27      .attr("transform", "translate(100,0)")
28      .attr("x", 50)
29      .attr("y", 50)
30      .attr("font-size", "24px")
31      .text("Stock Price")
32
33    var x = d3.scaleBand().range([0, width]).padding(0.4);
34    var y = d3.scaleLinear().range([height, 0]);
35    var g = svg.append("g")
36      .attr("transform", "translate(" + 100 + "," + 100 + ")");
37
38  d3.csv("https://raw.githubusercontent.com/HusainMehdi/DataVis/main/csvfile.csv").then(function (data) {
39    x.domain(data.map(function (d) { return d.year; }));
40    y.domain([0, d3.max(data, function (d) { return d.value; })]);
41    g.append("g")
42      .attr("transform", "translate(0," + height + ")")
43      .call(d3.axisBottom(x))
44      .append("text")
45      .attr("y", height - 250)
46      .attr("x", width - 100)
47      .attr("text-anchor", "end")
48      .attr("stroke", "black")
49      .text("Year");
50    g.append("g")
51      .call(d3.axisLeft(y).tickFormat(function (d) {
52        return "$" + d;
53      }).ticks(10))
54      .append("text")
55      .attr("transform", "rotate(-90)")
56      .attr("y", 6)
```

```

57     .attr("dy", "-5.1em")
58     .attr("text-anchor", "end")
59     .attr("stroke", "black")
60     .text("Stock Price");
61   g.selectAll(".bar")
62     .data(data)
63     .enter().append("rect")
64     .attr("class", "bar")
65     // .on(... ) - call mouse events here...
66
67     .on("mouseover", function onMouseOver(d, i) {
68       d3.select(this).attr('class', 'highlight');
69       d3.select(this)
70         .transition() // adds animation
71         .duration(400)
72         .attr('width', x.bandwidth() + 5)
73         .attr("y", function (d) { return y(d.value) - 10; })
74         .attr("height", function (d) { return height - y(d.value) + 10; });
75
76
77       g.append("text")
78         .attr('class', 'val')
79         .attr('x', function (d) {
80           return x(i.year);
81         })
82         .attr('y', function (d) {
83           return y(i.value) - 15;
84         })
85         .text(function (d) { return '$' + i.value; }); // Value of the text
86     })
87
88     .on("mouseout", function onMouseOut(d, i) {
89       // use the text label class to remove label onmouseout
90       d3.select(this).attr('class', 'bar');
91       d3.select(this)
92         .transition() // adds animation
93         .duration(400)
94         .attr('width', x.bandwidth())
95         .attr("y", function (d) { return y(i.value); })
96         .attr("height", function (d) { return height - y(i.value); });
97       d3.selectAll('.val')
98         .remove()
99     })
100
101
102     .attr("x", function (d) { return x(d.year); })
103     .attr("y", function (d) { return y(d.value); })
104     .attr("width", x.bandwidth())
105     .transition()
106     .ease(d3.easeLinear)
107     .duration(400)
108     .delay(function (d, i) {
109       return i * 50;
110     })
111     .attr("height", function (d) { return height - y(d.value); });

```

EX17

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5
6      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
7
8      <svg width="600" height="500"></svg>
9
10     <script>
11         var svg = d3.select("svg");
12         var margin = 200;
13         var width = svg.attr("width") - margin;
14         var height = svg.attr("height") - margin;
15
16         var myColor1 = d3.scaleLinear().domain([1, 10])
17             .range(["steelblue", "indianred"]);
18
19         var myColor2 = d3.scaleLinear().domain([1, 10])
20             .range(["darkblue", "darkred"]);
21
22         var values = [];
23
24         svg.append("text")
25             .attr("transform", "translate(100,0)")
26             .attr("x", 50)
27             .attr("y", 50)
28             .attr("font-size", "24px")
29             .text("Stock Price");
30
31         var x = d3.scaleBand().range([0, width]).padding(0.4);
32         var y = d3.scaleLinear().range([height, 0]);
33         var g = svg.append("g")
34             .attr("transform", "translate(" + 100 + "," + 100 + ")");
35
36         d3.csv("https://raw.githubusercontent.com/HusainMehdi/DataVis/main/csvfile.csv").then(function (data) {
37
38             for (var i = 0; i < data.length; i++) {
39                 values.push(data[i].value);
40             }
41
42             var scale = d3.scaleLinear()
43                 .domain([d3.min(values), d3.max(values)])
44                 .range([0, 10]);
45
46             x.domain(data.map(function (d) { return d.year; }));
47             y.domain([0, d3.max(data, function (d) { return d.value; })]);
48             g.append("g")
49                 .attr("transform", "translate(0," + height + ")")
50                 .call(d3.axisBottom(x))
51                 .append("text")
52                 .attr("y", height - 250)
53                 .attr("x", width - 100)
54                 .attr("text-anchor", "end")
55                 .attr("stroke", "black")
56                 .text("Year");

```

```

56           .text('Year');
57       g.append("g")
58           .call(d3.axisLeft(y).tickFormat(function (d) {
59               return "$" + d;
60           }).ticks(10))
61           .append("text")
62           .attr("transform", "rotate(-90)")
63           .attr("y", 6)
64           .attr("dy", "-5.1em")
65           .attr("text-anchor", "end")
66           .attr("stroke", "black")
67           .text("Stock Price");
68   g.selectAll(".bar")
69       .data(data)
70       .enter().append("rect")
71           .attr("class", "bar")
72           .style("fill", function (d) {
73               return myColor1(scale(d.value));
74           })
75       // .on(... ) - call mouse events here...
76
77       .on("mouseover", function onMouseOver(d, i) {
78           d3.select(this).attr('class', 'highlight');
79           d3.select(this)
80               .transition() // adds animation
81               .duration(400)
82               .attr('width', x.bandwidth() + 5)
83               .attr("y", function (d) { return y(d.value) - 10; })
84               .attr("height", function (d) { return height - y(d.value) + 10; })
85               .style("fill", function (d) {
86                   return myColor2(scale(d.value))
87               });
88
89
90           g.append("text")
91               .attr('class', 'val')
92               .attr('x', function (d) {
93                   return x(i.year);
94               })
95               .attr('y', function (d) {
96                   return y(i.value) - 15;
97               })
98               .text(function (d) { return '$' + i.value; }); // Value of the text
99       });
100
101      .on("mouseout", function onMouseOut(d, i) {
102          // use the text label class to remove label onmouseout
103          d3.select(this).attr('class', 'bar');
104          d3.select(this)
105              .transition() // adds animation
106              .duration(400)
107              .attr('width', x.bandwidth())
108              .attr("y", function (d) { return y(i.value); })
109              .attr("height", function (d) { return height - y(i.value); })
110              .style("fill", function (d) {
111                  return myColor1(scale(d.value));

```

```
112     });
113     d3.selectAll('.val')
114       .remove()
115   })
116
117
118   .attr("x", function (d) { return x(d.year); })
119   .attr("y", function (d) { return y(d.value); })
120   .attr("width", x.bandwidth())
121   .transition()
122   .ease(d3.easeLinear)
123   .duration(400)
124   .delay(function (d, i) {
125     return i * 50;
126   })
127   .attr("height", function (d) { return height - y(d.value); });
128 });
129
130
131 </script>
132 </body>
```

EX18

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5
6      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
7
8      <!-- Add buttons -->
9      <button onclick="update(data1)">Variable 1</button>
10     <button onclick="update(data2)">Variable 2</button>
11     <button onclick="update(data3)">Variable 3</button>
12
13     <script>
14         // create 2 data_set
15         const data1 = [
16             { group: "A", value: 5 },
17             { group: "B", value: 20 },
18             { group: "C", value: 9 }
19         ];
20         const data2 = [
21             { group: "A", value: 10 },
22             { group: "B", value: 2 },
23             { group: "C", value: 22 }
24         ];
25         const data3 = [
26             { group: "A", value: 6 },
27             { group: "B", value: 12 },
28             { group: "C", value: 18 }
29         ];
30
31         // set the dimensions and margins of the graph
32         const margin = { top: 30, right: 30, bottom: 70, left: 60 };
33         const width = 460 - margin.left - margin.right;
34         const height = 400 - margin.top - margin.bottom;
35
36         // append the svg object to the body of the page
37         var svg = d3.select('body')
38             .append('div')
39             .append("svg")
40             .attr("width", width + margin.left + margin.right)
41             .attr("height", height + margin.top + margin.bottom)
42             .append("g")
43             .attr("transform",
44                 "translate(" + margin.left + "," + margin.top + ")");
45
46         // X axis
47         var x = d3.scaleBand()
48             .range([0, width])
49             .domain(data1.map(function (d) { return d.group; }))
50             .padding(0.2);
51         svg.append("g")
52             .attr("transform", "translate(0," + height + ")")
53             .call(d3.axisBottom(x))
54
55         // Add Y axis
56         var y = d3.scaleLinear()
```

```
57      .domain([0, 20])
58      .range([height, 0]);
59      svg.append("g")
60      .attr("class", "myYaxis")
61      .call(d3.axisLeft(y));
62
63      // A function that create / update the plot for a given variable:
64      function update(data) {
65          var u = svg.selectAll("rect")
66          .data(data)
67          u.enter()
68          .append("rect")
69          .merge(u)
70          .transition()
71          .duration(1000)
72          .attr("x", function (d) { return x(d.group); })
73          .attr("y", function (d) { return y(d.value); })
74          .attr("width", x.bandwidth())
75          .attr("height", function (d) { return height - y(d.value); })
76          .attr("fill", "#69b3a2")
77      }
78      // Initialize the plot with the first dataset
79      update(data1)
80  </script>
81 </body>
```

EX19

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5
6      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
7
8      <!-- Add buttons -->
9      <button onclick="update(data1, 'goldenrod')">Variable 1</button>
10     <button onclick="update(data2, 'lightgreen')">Variable 2</button>
11     <button onclick="update(data3, 'lightsalmon')">Variable 3</button>
12
13     <script>
14         // create 2 data_set
15         const data1 = [
16             { group: "A", value: 5 },
17             { group: "B", value: 20 },
18             { group: "C", value: 9 }
19         ];
20         const data2 = [
21             { group: "A", value: 10 },
22             { group: "B", value: 2 },
23             { group: "C", value: 22 }
24         ];
25         const data3 = [
26             { group: "A", value: 6 },
27             { group: "B", value: 12 },
28             { group: "C", value: 18 }
29         ];
30
31         // set the dimensions and margins of the graph
32         const margin = { top: 30, right: 30, bottom: 70, left: 60 };
33         const width = 460 - margin.left - margin.right;
34         const height = 400 - margin.top - margin.bottom;
35
36         // append the svg object to the body of the page
37         var svg = d3.select('body')
38             .append('div')
39             .append("svg")
40             .attr("width", width + margin.left + margin.right)
41             .attr("height", height + margin.top + margin.bottom)
42             .append("g")
43             .attr("transform",
44                 "translate(" + margin.left + "," + margin.top + ")");
45
46         // X axis
47         var x = d3.scaleBand()
48             .range([0, width])
49             .domain(data1.map(function (d) { return d.group; }))
50             .padding(0.2);
51         svg.append("g")
52             .attr("transform", "translate(0," + height + ")")
53             .call(d3.axisBottom(x))
54
```

```
55 // Add Y axis
56 var y = d3.scaleLinear()
57   .domain([0, 20])
58   .range([height, 0]);
59 svg.append("g")
60   .attr("class", "myYaxis")
61   .call(d3.axisLeft(y));
62
63 // A function that create / update the plot for a given variable:
64 function update(data, colour) {
65   var u = svg.selectAll("rect")
66     .data(data)
67   u.enter()
68     .append("rect")
69     .merge(u)
70     .transition()
71     .duration(1000)
72     .attr("x", function (d) { return x(d.group); })
73     .attr("y", function (d) { return y(d.value); })
74     .attr("width", x.bandwidth())
75     .attr("height", function (d) { return height - y(d.value); })
76     .attr("fill", colour)
77 }
78 // Initialize the plot with the first dataset
79 update(data1, 'goldenrod')
80 </script>
81 </body>
```

EX20

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5
6      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
7
8      <!-- Add buttons -->
9      <button onclick="update(data1, 'goldenrod')">Variable 1</button>
10     <button onclick="update(data2, 'lightgreen')">Variable 2</button>
11     <button onclick="update(data3, 'lightsalmon')">Variable 3</button>
12
13  <script>
14      // create 2 data_set
15  const data1 = [
16      { group: "A", value: 5 },
17      { group: "B", value: 20 },
18      { group: "C", value: 9 }
19  ];
20  const data2 = [
21      { group: "A", value: 10 },
22      { group: "B", value: 2 },
23      { group: "C", value: 22 }
24  ];
25  const data3 = [
26      { group: "A", value: 6 },
27      { group: "B", value: 12 },
28      { group: "C", value: 18 }
29  ];
30
31      // set the dimensions and margins of the graph
32  const margin = { top: 30, right: 30, bottom: 70, left: 60 };
33  const width = 460 - margin.left - margin.right;
34  const height = 400 - margin.top - margin.bottom;
35
36      // append the svg object to the body of the page
37  var svg = d3.select('body')
38      .append('div')
39      .append("svg")
40      .attr("width", width + margin.left + margin.right)
41      .attr("height", height + margin.top + margin.bottom)
42      .append("g")
43      .attr("transform",
44          | "translate(" + margin.left + "," + margin.top + ")");
45
46      // X axis
47  var x = d3.scaleBand()
48      .range([0, width])
49      .domain(data1.map(function (d) { return d.group; }))
50      .padding(0.2);
51  svg.append("g")
52      .attr("transform", "translate(0," + height + ")")
53      .call(d3.axisBottom(x))
54
55      // Add Y axis
56  var y = d3.scaleLinear()
```

```

56     var y = d3.scaleLinear()
57         .domain([0, 20])
58         .range([height, 0]);
59     svg.append("g")
60         .attr("class", "myYaxis")
61         .call(d3.axisLeft(y));
62
63 // A function that create / update the plot for a given variable:
64 function update(data, colour) {
65     var u = svg.selectAll("rect")
66         .data(data)
67
68     u.enter()
69         .append("rect")
70         .merge(u)
71         .attr("x", function (d) {
72             console.log(d.value);
73             return 1;
74         })
75         .attr("y", "1")
76         .on("mouseover", function onMouseOver(d, i) {
77             console.log("jhg");
78
79             svg.append("text")
80                 .attr('class', 'val')
81                 .attr('x', function (d) { return x(i.group); })
82                 .attr('y', function (d) { console.log(y(i.value)); return y(i.value); })
83                 .text(function (d) {
84                     console.log(i.value);
85                     return i.value;
86                 });
87         })
88         .on("mouseout", function onMouseOut(d, i) {
89             console.log("hfgcvjbjbchjk");
90             svg.selectAll('.val')
91                 .remove();
92         })
93         .transition()
94         .duration(1000)
95         .attr("x", function (d) { return x(d.group); })
96         .attr("y", function (d) { return y(d.value); })
97         .attr("width", x.bandwidth())
98         .attr("height", function (d) { return height - y(d.value); })
99         .attr("fill", colour)
100        .attr("class", "bar")
101    }
102
103 // Initialize the plot with the first dataset
104 update(data1, 'goldenrod')
105 </script>
106 </body>

```

EX21

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4 <body>
5
6   <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
7
8   <!-- Add buttons -->
9   <button onclick="update(data1, 'goldenrod')">Variable 1</button>
10  <button onclick="update(data2, 'lightgreen')">Variable 2</button>
11  <button onclick="update(data3, 'lightsalmon')">Variable 3</button>
12
13 <script>
14   // create 2 data_set
15   const data1 = [
16     { group: "A", value: 5 },
17     { group: "B", value: 20 },
18     { group: "C", value: 9 }
19   ];
20   const data2 = [
21     { group: "A", value: 10 },
22     { group: "B", value: 2 },
23     { group: "C", value: 22 }
24   ];
25   const data3 = [
26     { group: "A", value: 6 },
27     { group: "B", value: 12 },
28     { group: "C", value: 18 }
29   ];
30
31   // set the dimensions and margins of the graph
32   const margin = { top: 30, right: 30, bottom: 70, left: 60 };
33   const width = 460 - margin.left - margin.right;
34   const height = 400 - margin.top - margin.bottom;
35
36   // append the svg object to the body of the page
37   var svg = d3.select('body')
38     .append('div')
39     .append("svg")
40     .attr("width", width + margin.left + margin.right)
41     .attr("height", height + margin.top + margin.bottom)
42     .append("g")
43     .attr("transform",
44       "translate(" + margin.left + "," + margin.top + ")");
45
46   // X axis
47   var x = d3.scaleBand()
48     .range([0, width])
49     .domain(data1.map(function (d) { return d.group; }))
50     .padding(0.2);
51   svg.append("g")
52     .attr("transform", "translate(0," + height + ")")
53     .call(d3.axisBottom(x));
54   svg.append("g")
55     .attr("transform", "translate(0," + 0 + ")")
56     .call(d3.axisTop(x));
```

```
57
58     // Add Y axis
59     var y = d3.scaleLinear()
60         .domain([0, 20])
61         .range([height, 0]);
62     svg.append("g")
63         .attr("class", "myYaxis")
64         .call(d3.axisLeft(y));
65     svg.append("g")
66         .attr("class", "myYaxis")
67         .attr("transform", "translate(" + width + ", 0)")
68         .call(d3.axisRight(y));
69
70     // A function that create / update the plot for a given variable:
71     function update(data, colour) {
72         var u = svg.selectAll("rect")
73             .data(data)
74         u.enter()
75             .append("rect")
76             .merge(u)
77             .transition()
78             .duration(1000)
79             .attr("x", function (d) { return x(d.group); })
80             .attr("y", function (d) { return y(d.value); })
81             .attr("width", x.bandwidth())
82             .attr("height", function (d) { return height - y(d.value); })
83             .attr("fill", colour)
84     }
85     // Initialize the plot with the first dataset
86     update(data1, 'goldenrod')
87
88 </script>
89 </body>
```

EX22

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5
6      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
7
8      <!-- Add buttons -->
9      <button onclick="update(data1, 'goldenrod')">Variable 1</button>
10     <button onclick="update(data2, 'lightgreen')">Variable 2</button>
11     <button onclick="update(data3, 'lightsalmon')">Variable 3</button>
12
13     <script>
14         // create 2 data_set
15         const data1 = [
16             { group: "A", value: 5 },
17             { group: "B", value: 20 },
18             { group: "C", value: 9 }
19         ];
20         const data2 = [
21             { group: "A", value: 10 },
22             { group: "B", value: 2 },
23             { group: "C", value: 22 },
24             { group: "D", value: 6 }
25         ];
26         const data3 = [
27             { group: "A", value: 6 },
28             { group: "B", value: 12 },
29             { group: "C", value: 18 }
30         ];
31
32         // set the dimensions and margins of the graph
33         const margin = { top: 30, right: 30, bottom: 70, left: 60 };
34         const width = 460 - margin.left - margin.right;
35         const height = 400 - margin.top - margin.bottom;
36
37         // append the svg object to the body of the page
38         var svg = d3.select('body')
39             .append('div')
40             .append("svg")
41             .attr("width", width + margin.left + margin.right)
42             .attr("height", height + margin.top + margin.bottom)
43             .append("g")
44             .attr("transform",
45                 "translate(" + margin.left + "," + margin.top + ")");
46
47
48
49         // A function that create / update the plot for a given variable:
50         function update(data, colour) {
51             var u = svg.selectAll("rect")
52                 .data(data)
53
54                 svg.selectAll("g").remove();
55
56

```

```
57 // X axis
58 var x = d3.scaleBand()
59   .range([0, width])
60   .domain(data.map(function (d) { return d.group; }))
61   .padding(0.2);
62 svg.append("g")
63   .attr("transform", "translate(0," + height + ")")
64   .call(d3.axisBottom(x))
65 svg.append("g")
66   .attr("transform", "translate(0," + 0 + ")")
67   .call(d3.axisTop(x))
68
69 // Add Y axis
70 var y = d3.scaleLinear()
71   .domain([0, 20])
72   .range([height, 0]);
73 svg.append("g")
74   .attr("class", "myYaxis")
75   .call(d3.axisLeft(y));
76 svg.append("g")
77   .attr("class", "myYaxis")
78   .attr("transform", "translate(" + width + ", 0)")
79   .call(d3.axisRight(y));
80
81 u.exit().remove();
82
83 u.enter()
84   .append("rect")
85   .merge(u)
86   .transition()
87   .duration(1000)
88   .attr("x", function (d) { return x(d.group); })
89   .attr("y", function (d) { return y(d.value); })
90   .attr("width", x.bandwidth())
91   .attr("height", function (d) { return height - y(d.value); })
92   .attr("fill", colour)
93
94 }
95
96 // Initialize the plot with the first dataset
97 update(data1, 'goldenrod')
98 </script>
99 </body>
```

EX23

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5
6      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
7
8      <!-- Add buttons -->
9      <button onclick="update(data1, 'goldenrod')">Variable 1</button>
10     <button onclick="update(data2, 'lightgreen')">Variable 2</button>
11
12
13     <script>
14         // create 2 data_set
15 // Set Dimensions
16 const xSize = 600; const ySize = 600;
17 const margin = 40;
18 const xMax = xSize - margin * 2;
19 const yMax = ySize - margin * 2;
20
21     // Create Random Points
22 const numPoints = 100;
23 const data1 = [];
24 for (let i = 0; i < numPoints; i++) { data1.push({ x: i / 100, y: Math.sin(6.2 * i / 100) }); }
25
26 const data2 = [];
27 for (let i = 0; i < numPoints; i++) { data2.push({ x: i / 100, y: Math.cos(6.2 * i / 100) }); }
28
29
30     // Get the 'limits' of the data – the full extent (mins and max)
31     // so the plotted data fits perfectly
32 const xExtent = d3.extent(data1, d => { return d.x });
33 const yExtent = d3.extent(data1, d => { return d.y });
34
35     // Append SVG Object to the Page
36 const svg = d3.select("body")
37     .append("svg")
38     .attr('width', xSize)
39     .attr('height', ySize)
40     .append("g")
41     .attr("transform", "translate(" + margin + "," + margin + ")");
42
43     // X Axis
44 const x = d3.scaleLinear()
45     .domain([xExtent[0], xExtent[1]])
46     .range([0, xMax]);
47
48     // bottom
49     svg.append("g")
50     .attr("transform", "translate(0," + yMax + ")")
51     .call(d3.axisBottom(x))
52     .attr('color', 'green'); // make bottom axis green
53
54     // top
55     svg.append("g")
56     .call(d3.axisTop(x));
```

```
56     .call(d3.axisTop(x));
57
58     // Y Axis
59     const y = d3.scaleLinear()
60         .domain([yExtent[0], yExtent[1]])
61         .range([yMax, 0]);
62
63     // left y axis
64     svg.append("g")
65         .call(d3.axisLeft(y));
66
67     // right y axis
68     svg.append("g")
69         .attr("transform", `translate(${yMax},0`)
70         .call(d3.axisRight(y));
71
72
73
74     // A function that create / update the plot for a given variable:
75     function update(data, colour) {
76         var u = svg.selectAll("rect")
77             .data(data)
78
79
80         svg.selectAll(".line").remove();
81
82
83         // Add the line
84         svg.append("path")
85             .datum(data)
86             .attr("fill", "none")
87             .attr("class", "line")
88             .attr("stroke", "steelblue")
89             .attr("stroke-width", 1.5)
90             .attr("d", d3.line()
91                 .x(function (d) { return x(d.x) })
92                 .y(function (d) { return y(d.y) })
93             );
94
95
96
97     }
98     // Initialize the plot with the first dataset
99     update(data1, 'goldenrod')
100    </script>
101  </body>
```

EX24

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <p>1) Script tag is missing "https:", resulting in error: "lab2-24.html:8 Uncaught ReferenceError: d3 is not defined"</p>
6      <p>2) Afterwards the output is a list containing the values 20% between the values in the first and second lists of function "intr"</p>
7      <script type='text/javascript' src="https://d3js.org/d3.v7.min.js"></script>
8      <script>
9          let intr = d3.interpolate( [20, 40, 4], [1, 12, 10] )
10         console.log("Type of returned function is: ", typeof (intr) );
11         console.log( intr(0.2) )
12     </script>
13 </body>
```

EX25

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6
7
8      <p>The returned value is 'rgb(128, 64, 0)', 0.5 the way between green [rgb(0, 128, 0)], and red [rgb(256, 0, 0)] </p>
9
10
11     <script>
12         let cc = d3.interpolate('red', 'green')
13         console.log( cc(0.5) );
14
15         d3.select("p").style("color",function(){return cc(0.5)})
16
17     </script>
18 </body>
```

EX26

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6
7
8      <p></p>
9
10
11     <script>
12         let intrDate = d3.interpolateDate(new Date("01/01/2000"), new Date("02/22/2022"))
13         console.log( intrDate(0.5) );
14
15         d3.select("p").style("color",function(){return intrDate(0.5)})
16         .text("The day between 01/01/2000 and 02/22/2022 is " + intrDate(0.5))
17
18     </script>
19 </body>
```

EX27

```
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6
7  |     <button onclick="update(dataset3.apples)">Variable 1</button>
8  |     <button onclick="update(dataset2.apples)">Variable 2</button>
9
10 <script>
11     var dataset = {
12         apples: [5345, 2879, 1997, 2437, 4045],
13     };
14     var dataset2 = {
15         apples: [23, 56, 2, 67, 30],
16     };
17     var dataset3 = {
18         apples: [1, 2, 3, 4, 5],
19     };
20
21     var width = 460,
22         height = 300,
23         radius = Math.min(width, height) / 2;
24
25     var color = d3.scaleOrdinal().range(d3.schemeSet3);
26
27     var pie = d3.pie()
28         .sort(null);
29
30     var arc = d3.arc()
31         .innerRadius(radius - 100)
32         .outerRadius(radius - 50);
33
34     var svg = d3.select("body").append("svg")
35         .attr("width", width)
36         .attr("height", height)
37         .append("g")
38         .attr("transform", "translate(" + width / 2 + "," + height / 2 + ")");
39
40     function update(data) {
41         svg.selectAll("path").remove();
42
43         var path = svg.selectAll("path")
44             .data(data)
45             .enter()
46             .append("path")
47             .attr("fill", function (d, i) { return color(i); })
48             .attr("d", arc)
49             .transition()
50             .duration(1000)
51             .attrTween("d", function (d) {
52                 var i = d3.interpolate(d.endAngle, d.startAngle);
53                 return function (t) {
54                     d.startAngle = i(t);
55                     return arc(d);
56                 }
57             });
58     }
59     update(dataset2.apples);
60   </script>
61 </body>
```

EX28

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7          var width = 400, height = 400;
8
9          var myColor = d3.scaleSequential().domain([0, 25]).interpolator(d3.interpolateViridis);
10
11         // setup svg
12         d3.select('body').append('svg').attr('width', width).attr('height', height);
13
14         // generate some random data
15         var numNodes = 100;
16         var nodes = d3.range(numNodes).map(function (d) {
17             return { radius: Math.random() * 25 }
18         })
19
20         console.log(nodes[0]);
21
22
23         var scale = d3.scaleLinear()
24             .domain([d3.min(nodes), d3.max(nodes)])
25             .range([0, 10]);
26
27         var simulation = d3.forceSimulation(nodes)
28             .force('charge', d3.forceManyBody().strength(5))
29             .force('center', d3.forceCenter(width / 2, height / 2))
30             .force('collision', d3.forceCollide().radius(function (d) {
31                 return d.radius
32             }))
33             .on('tick', ticked);
34
35         function ticked() {
36             var u = d3.select('svg')
37                 .selectAll('circle')
38                 .data(nodes)
39                 .join('circle')
40                 .attr('fill', function (d) {
41                     return myColor(d.radius)
42                 })
43                 .attr('r', function (d) {
44                     return d.radius
45                 })
46                 .attr('cx', function (d) {
47                     return d.x
48                 })
49                 .attr('cy', function (d) {
50                     return d.y
51                 })
52         }
53         console.log('ready..');
54
55
56     </script>
57 </body>
```

EX29

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7          var width = 400, height = 400;
8
9          var myColor = d3.scaleSequential().domain([0, 25]).interpolator(d3.interpolateViridis);
10
11         // setup svg
12         d3.select('body').append('svg').attr('width', width).attr('height', height);
13
14         // generate some random data
15         var numNodes = 100;
16         // var nodes = d3.range(numNodes).map(function (d) {
17         //     return { radius: Math.random() * 25 }
18         // })
19
20         const nodes = [
21             { radius: 5 },
22             { radius: 25 },
23             { radius: 9 },
24             { radius: 1 },
25             { radius: 2 },
26             { radius: 3 },
27             { radius: 4 },
28             { radius: 5 },
29             { radius: 6 },
30             { radius: 7 },
31             { radius: 8 },
32             { radius: 9 },
33             { radius: 10 },
34             { radius: 11 },
35             { radius: 12 },
36             { radius: 13 },
37             { radius: 14 },
38             { radius: 15 },
39             { radius: 16 },
40             { radius: 17 },
41             { radius: 18 },
42             { radius: 19 },
43             { radius: 20 },
44             { radius: 21 },
45             { radius: 22 },
46             { radius: 23 },
47             { radius: 24 },
48             { radius: 25 }
49         ];
50
51         console.log(nodes)
52
53
54         var scale = d3.scaleLinear()
55             .domain([d3.min(nodes), d3.max(nodes)])
56             .range([0, 10]);
57
58         var simulation = d3.forceSimulation(nodes)
59             .force('charge', d3.forceManyBody().strength(5))
```

```
60      .force('center', d3.forceCenter(width / 2, height / 2))
61      .force('collision', d3.forceCollide().radius(function (d) {
62          return d.radius
63      }))
64      .on('tick', ticked);
65
66  function ticked() {
67      var u = d3.select('svg')
68          .selectAll('circle')
69          .data(nodes)
70          .join('circle')
71          .attr('fill', function (d) {
72              return myColor(d.radius)
73          })
74          .attr('r', function (d) {
75              return d.radius
76          })
77          .attr('cx', function (d) {
78              return d.x
79          })
80          .attr('cy', function (d) {
81              return d.y
82          })
83      }
84      console.log('ready..');
85
86
87  </script>
88 </body>
```

EX30

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7
8          var div = d3.select("body").append("div")
9              .attr("class", "tooltip")
10             .style("opacity", 0);
11
12         var width = 400, height = 400;
13
14         var myColor = d3.scaleSequential().domain([0, 25]).interpolator(d3.interpolateViridis);
15
16         // setup svg
17         d3.select('body').append('svg').attr('width', width).attr('height', height);
18
19
20         // generate some random data
21         var numNodes = 100;
22         // var nodes = d3.range(numNodes).map(function (d) {
23         //     return { radius: Math.random() * 25 }
24         // })
25
26         const nodes = [
27             { radius: 5 },
28             { radius: 25 },
29             { radius: 9 },
30             { radius: 1 },
31             { radius: 2 },
32             { radius: 3 },
33             { radius: 4 },
34             { radius: 5 },
35             { radius: 6 },
36             { radius: 7 },
37             { radius: 8 },
38             { radius: 9 },
39             { radius: 10 },
40             { radius: 11 },
41             { radius: 12 },
42             { radius: 13 },
43             { radius: 14 },
44             { radius: 15 },
45             { radius: 16 },
46             { radius: 17 },
47             { radius: 18 },
48             { radius: 19 },
49             { radius: 20 },
50             { radius: 21 },
51             { radius: 22 },
52             { radius: 23 },
53             { radius: 24 },
54             { radius: 25 }
55         ];
56
57         console.log(nodes)
58
59     </script>
```

```
60      var scale = d3.scaleLinear()
61          .domain([d3.min(nodes), d3.max(nodes)])
62          .range([0, 10]);
63
64      var simulation = d3.forceSimulation(nodes)
65          .force('charge', d3.forceManyBody().strength(5))
66          .force('center', d3.forceCenter(width / 2, height / 2))
67          .force('collision', d3.forceCollide()).radius(function (d) {
68              return d.radius
69          })
70          .on('tick', ticked);
71
72      function ticked() {
73          var u = d3.select('svg')
74              .selectAll('circle')
75              .data(nodes)
76              .join('circle')
77              .attr('fill', function (d) {
78                  return myColor(d.radius)
79              })
80              .attr('r', function (d) {
81                  return d.radius
82              })
83              .attr('cx', function (d) {
84                  return d.x
85              })
86              .attr('cy', function (d) {
87                  return d.y
88              })
89              .on('mousemove', (event, d) => {
90                  console.log(d)
91                  var coords = d3.pointer(event);
92                  console.log(coords[0], coords[1]) // log the mouse x,y position
93                  div.transition()
94                      .duration(200)
95                      .style("opacity", .9);
96                  div.html("Radius: " + d.radius)
97                      .style("left", (coords[0]) + "px")
98                      .style("top", (coords[1]) + "px");
99              });
100      }
101      console.log('ready..');
102
103
104  </script>
105  <style>
106      div.tooltip {
107          position: absolute;
108          text-align: center;
109          width: 60px;
110          height: 28px;
111          padding: 2px;
112          font: 12px sans-serif;
113          background: #lightsteelblue;
114          border: 0px;
115          border-radius: 8px;
116          pointer-events: none;
117      }
118  </style>
119</body>
```

EX31

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7
8          var div = d3.select("body").append("div")
9              .attr("class", "tooltip")
10             .style("opacity", 0);
11
12         var width = 400, height = 400;
13
14         var myColor = d3.scaleSequential().domain([0, 25]).interpolator(d3.interpolateViridis);
15
16         var myColor2 = d3.scaleLinear().domain([1, 25])
17             .range(["pink", "goldenrod"]);
18
19         // setup svg
20         d3.select('body').append('svg').attr('width', width).attr('height', height);
21
22
23         // generate some random data
24         var numNodes = 100;
25         // var nodes = d3.range(numNodes).map(function (d) {
26         //     return { radius: Math.random() * 25 }
27         // })
28
29         const nodes = [
30             { radius: 5 },
31             { radius: 25 },
32             { radius: 9 },
33             { radius: 1 },
34             { radius: 2 },
35             { radius: 3 },
36             { radius: 4 },
37             { radius: 5 },
38             { radius: 6 },
39             { radius: 7 },
40             { radius: 8 },
41             { radius: 9 },
42             { radius: 10 },
43             { radius: 11 },
44             { radius: 12 },
45             { radius: 13 },
46             { radius: 14 },
47             { radius: 15 },
48             { radius: 16 },
49             { radius: 17 },
50             { radius: 18 },
51             { radius: 19 },
52             { radius: 20 },
53             { radius: 21 },
54             { radius: 22 },
55             { radius: 23 },
56             { radius: 24 },
57             { radius: 25 }
58         ];
59     
```

```
63      var scale = d3.scaleLinear()
64          .domain([d3.min(nodes), d3.max(nodes)])
65          .range([0, 10]);
66
67      var simulation = d3.forceSimulation(nodes)
68          .force('charge', d3.forceManyBody().strength(5))
69          .force('center', d3.forceCenter(width / 2, height / 2))
70          .force('collision', d3.forceCollide().radius(function (d) {
71              return d.radius
72            }))
73          .on('tick', ticked);
74
75      function ticked() {
76          var u = d3.select('svg')
77              .selectAll('circle')
78              .data(nodes)
79              .join('circle')
80              .attr('fill', function (d) {
81                  return myColor(d.radius)
82                })
83              .attr('r', function (d) {
84                  return d.radius
85                })
86              .attr('cx', function (d) {
87                  return d.x
88                })
89              .attr('cy', function (d) {
90                  return d.y
91                })
92              .on('mousemove', (event, d) => {
93                  console.log(d)
94                  var coords = d3.pointer(event);
95                  console.log(coords[0], coords[1]) // log the mouse x,y position
96                  div.transition()
97                      .duration(200)
98                      .style("opacity", .9);
99                  div.html("Radius: " + d.radius)
100                     .style("left", (coords[0]) + "px")
101                     .style("top", (coords[1]) + "px");
102              })
103              .on("mouseover", function onMouseOver(d, i) {
104                  d3.select(this).attr('class', 'highlight');
105                  d3.select(this)
106                      .transition() // adds animation
107                      .duration(400)
108                      .style("fill", myColor2(i.radius));
109              })
110              .on("mouseout", function onMouseOut(d, i) {
111                  d3.select(this).attr('class', 'highlight');
112                  d3.select(this)
113                      .transition() // adds animation
114                      .duration(400)
115                      .style("fill", myColor(i.radius));
116              });
117      }
118      console.log('ready..');
119
120
121  </script>
```

```
122 <style>
123     div.tooltip {
124         position: absolute;
125         text-align: center;
126         width: 60px;
127         height: 28px;
128         padding: 2px;
129         font: 12px sans-serif;
130         background: lightsteelblue;
131         border: 0px;
132         border-radius: 8px;
133         pointer-events: none;
134     }
135 </style>
136 </body>
```

EX32

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7
8          var div = d3.select("body").append("div")
9              .attr("class", "tooltip")
10             .style("opacity", 0);
11
12     var width = 400, height = 400;
13
14     var myColor = d3.scaleSequential().domain([0, 25]).interpolator(d3.interpolateViridis);
15
16     var myColor2 = d3.scaleLinear().domain([1, 25])
17         .range(["pink", "goldenrod"]);
18
19     // setup svg
20     d3.select('body').append('svg').attr('width', width).attr('height', height);
21
22
23     // generate some random data
24     var numNodes = 100;
25     // var nodes = d3.range(numNodes).map(function (d) {
26     //     return { radius: Math.random() * 25 }
27     // })
28
29     const nodes = [
30         { radius: 5 },
31         { radius: 25 },
32         { radius: 9 },
33         { radius: 1 },
34         { radius: 2 },
35         { radius: 3 },
36         { radius: 4 },
37         { radius: 5 },
38         { radius: 6 },
39         { radius: 7 },
40         { radius: 8 },
41         { radius: 9 },
42         { radius: 10 },
43         { radius: 11 },
44         { radius: 12 },
45         { radius: 13 },
46         { radius: 14 },
47         { radius: 15 },
48         { radius: 16 },
49         { radius: 17 },
50         { radius: 18 },
51         { radius: 19 },
52         { radius: 20 },
53         { radius: 21 },
54         { radius: 22 },
55         { radius: 23 },
56         { radius: 24 },
57         { radius: 25 }
58     ];
```

```

60      var links = [
61        { source: nodes[1], target: nodes[26]},
62        { source: nodes[27], target: nodes[26]},
63        { source: nodes[1], target: nodes[27]},
64
65        { source: nodes[0], target: nodes[2]},
66        { source: nodes[2], target: nodes[3]},
67        { source: nodes[3], target: nodes[4]},
68        { source: nodes[4], target: nodes[5]},
69        { source: nodes[5], target: nodes[6]},
70        { source: nodes[6], target: nodes[7]},
71        { source: nodes[7], target: nodes[0]},
72        { source: nodes[0], target: nodes[8]},
73        { source: nodes[8], target: nodes[9]},
74        { source: nodes[10], target: nodes[11]},
75        { source: nodes[11], target: nodes[12]},
76
77        { source: nodes[2], target: nodes[6]},
78        { source: nodes[4], target: nodes[0]}
79    ]
80
81    console.log(nodes)
82
83
84    var scale = d3.scaleLinear()
85      .domain([d3.min(nodes), d3.max(nodes)])
86      .range([0, 10]);
87
88    var simulation = d3.forceSimulation(nodes)
89      .force('charge', d3.forceManyBody().strength(5))
90      .force('center', d3.forceCenter(width / 2, height / 2))
91      .force('collision', d3.forceCollide().radius(function (d) {
92        return d.radius
93      }))
94      .force('link', d3.forceLink().links(links).distance(10))
95      .on('tick', ticked);
96
97    function ticked() {
98      var u = d3.select('svg')
99        .selectAll('circle')
100        .data(nodes)
101        .join('circle')
102        .attr('fill', function (d) {
103          return myColor(d.radius)
104        })
105        .attr('r', function (d) {
106          return d.radius
107        })
108        .attr('cx', function (d) {
109          return d.x
110        })
111        .attr('cy', function (d) {
112          return d.y
113        })
114        .on('mousemove', (event, d) => {
115          console.log(d)
116          var coords = d3.pointer(event);
117          console.log(coords[0], coords[1]) // log the mouse x,y position
118          div.transition()

```

```
119      .duration(200)
120      .style("opacity", .9);
121      div.html("Radius: " + d.radius)
122      .style("left", (coords[0]) + "px")
123      .style("top", (coords[1]) + "px");
124    })
125    .on("mouseover", function onMouseOver(d, i) {
126      d3.select(this).attr('class', 'highlight');
127      d3.select(this)
128        .transition() // adds animation
129        .duration(400)
130        .style("fill", myColor2(i.radius));
131    })
132    .on("mouseout", function onMouseOut(d, i) {
133      d3.select(this).attr('class', 'highlight');
134      d3.select(this)
135        .transition() // adds animation
136        .duration(400)
137        .style("fill", myColor(i.radius));
138    });
139  }
140  console.log('ready..');
141
142
143 </script>
144 <style>
145   div.tooltip {
146     position: absolute;
147     text-align: center;
148     width: 60px;
149     height: 28px;
150     padding: 2px;
151     font: 12px sans-serif;
152     background: #lightsteelblue;
153     border: 0px;
154     border-radius: 8px;
155     pointer-events: none;
156   }
157 </style>
158 </body>
```