

Husain Mehdi

F20DV

Lab 1

Demonstrated 4 Feb 2022 to Amit Parekh

ex1

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  | <script src='https://d3js.org/d3.v7.min.js'></script>
5  </head>
6  <body>
7  |   7.3.0
8  <script>
9  | // write your d3 code here..
10 | console.log('d3.version:', d3.version);
11 </script>
12 </body>
13 </html>
```

ex2

```
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <script src='https://d3js.org/d3.v7.min.js'></script>
5      </head>
6
7      <body>
8          <script>
9              // write your d3 code here..
10             console.log('d3.version:', d3.version);
11
12         </script>
13
14         <!-- Exercise 2-->
15         <p>First paragraph</p>
16         <p>Second paragraph</p>
17         <script>
18             d3.select("p").style("color", "red")
19                 .style("font-size", "30px")
20                 .style("font-family", "Arial, Helvetica, sans-serif")
21                 .style("text-align", "center");
22
23         </script>
24
25
26     </body>
27 </html>
```

ex3

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <script src='https://d3js.org/d3.v7.min.js'></script>
5  |</head>
6
7  <body>
8  <script>
9  |   for (let i = 1; i < 11; i++) {
10 |
11 |       let newspan = d3.select("body").append('div');
12 |       newspan.text(i);
13 |
14 |       if (i < 6){
15 |           newspan.style("color", "red");
16 |       }
17 |       else{
18 |           newspan.style("color", "green");
19 |       }
20 |
21 |   }
22 |</script>
23 |</body>
24 |</html>
```

ex4

```
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <script src='https://d3js.org/d3.v7.min.js'></script>
5      </head>
6
7      <body>
8          <script>
9              for (let i = 1; i < 11; i++) {
10
11                  let newspan = d3.select("body").append('div');
12                  newspan.text(i);
13                  newspan.attr("id", 'p'+i);
14                  if (i < 6){
15                      newspan.style("color", "red");
16                  }
17                  else{
18                      newspan.style("color", "green");
19                  }
20
21                  d3.select('#p1').text("start");
22                  d3.select('#p1').style("color", "purple");
23
24                  d3.select('#p10').text("end");
25                  d3.select('#p10').style("color", "pink");
26
27          </script>
28      </body>
29
30  </html>
```

ex5

```
1 < MDN Reference
2 <html lang="en">
3   <head>
4     <script src='https://d3js.org/d3.v7.min.js'></script>
5   </head>
6
7   <body>
8   </body>
9
10  <script>
11    d3.select("body").append("div").text("Hello World!").style("color", "green");
12  </script>
13 </html>
```

ex6

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <script src='https://d3js.org/d3.v7.min.js'></script>
4
5  <body>
6    <div></div>
7    <div></div>
8    <div></div>
9
10 <script>
11   let otherdata = [{ name: 'test', val: 1, color: 'cyan'},
12   |   |   |   { name: 'other', val: 2, color: 'magenta'},
13   |   |   |   { name: 'b', val: 3, color: 'yellow'}];
14
15
16   let paragraph = d3.select("body")
17     .selectAll("div")
18     .data(otherdata)
19     .text(function (d, i) {
20       console.log("d.name: " + d.name);
21       console.log("d.val: " + d.val);
22       console.log("i: " + i);
23       console.log("this: " + this);
24
25       return 'cont:' + d.color; // return value is used to set the 'text'
26     });
27   </script>
28 </body>
29
30 </html>
```

ex7

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <script src='https://d3js.org/d3.v7.min.js'></script>
4
5  <body>
6    <div></div>
7    <div></div>
8    <div></div>
9    <div></div>
10
11 <script>
12   let num = [10, 50, 100, 200];
13   let paragraph = d3.select("body")
14     .selectAll("div")
15     .data(num)
16     .text(function (d, i) {
17       return 'cont:' + d; // return value is used to set the 'text'
18     })
19     .style("color", function (d, i) {
20       if (d >= 50 && d <= 100) {
21         return "red";
22       } else {
23         return "yellow";
24       }
25       return 'blue';
26     });
27   </script>
28 </body>
29
30 </html>
```

ex8

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <script src='https://d3js.org/d3.v7.min.js'></script>
4
5  <body>
6      <script>
7          var myData = ['a', 4, 1, 'b', 6, 2, 8, 9, 'z' ];
8
9          var span = d3.select("body")
10         .selectAll("span")
11         .data(myData)
12         .enter()
13         .append('span')
14         .text(function (d, i) {
15             return d;
16         })
17         .style("color", function(s){
18             if(isNaN(s)){
19                 return "blue";
20             }
21             return "green"
22         });
23     </script>
24 </body>
25
26 </html>
```

ex9

```

1   <!DOCTYPE html>
2   <html lang="en">
3
4   <body>
5     <script src='https://d3js.org/d3.v7.min.js'></script>
6     <script>
7       let titaniccsv = 'https://raw.githubusercontent.com/dsindy/kaggle-titanic/master/data/test.csv';
8     /*
9      PassengerId,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
10     892,3,"Kelly, Mr. James",male,34.5,0,0,330911,7.8292,,Q
11     893,3,"Wilkes, Mrs. James (Ellen Needs)",female,47,1,0,363272,7,,S
12     894,2,"Myles, Mr. Thomas Francis",male,62,0,0,240276,9.6875,,Q
13     895,3,"Wirz, Mr. Albert",male,27,0,0,315154,8.6625,,S
14   */
15
16   let mr = 0;
17   let mrs = 0;
18   let mas = 0;
19   let miss = 0;
20   let na = 0;
21   let male = 0;
22   let female = 0;
23
24   d3.csv(titaniccsv, function (data) {
25     console.log(data.Name);
26     if(data.Name.includes("Mr.")){
27       mr++;
28     }
29     else if(data.Name.includes("Mrs.")){
30       mrs++;
31     }
32     else if(data.Name.includes("Master")){
33       mas++;
34     }
35     else if(data.Name.includes("Miss")){
36       miss++;
37     }
38     else{
39       na++;
40       console.log('na');
41     }
42     if(data.Sex === 'male'){
43       male++;
44     }
45     else if(data.Sex === 'female'){
46       female++;
47     }
48     else{
49       console.log('na');
50     }
51
52   }).then(function(){
53     console.log("Mr:" + mr);
54     console.log("Mrs:" + mrs);
55     console.log("Master:" + mas);
56     console.log("Miss:" + miss);
57     console.log("Other: " + na);
58     console.log("Male: " + male);
59     console.log("Female: " + female);
60   })

```

ex10

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7
8          let heartfailurecsv = 'https://raw.githubusercontent.com/akmand/datasets/master/heart_failure.csv';
9
10         /*
11             age,anaemia,creatinine_phosphokinase,diabetes,ejection_fraction,high_blood_pressure,platelets,serum_creatini
12             ne,serum_sodium,sex,smoking,time,DEATH_EVENT
13             75,0,582,0,20,1,265000,1.9,130,1,0,4,1
14             55,0,7861,0,38,0,263358.03,1.1,136,1,0,6,1
15             65,0,146,0,20,0,162000,1.3,129,1,1,7,1
16             50,1,111,0,20,0,210000,1.9,137,1,0,7,1
17             */
18
19         var ageRanges = [0, 0, 0, 0]
20
21         d3.csv(heartfailurecsv, function (data) {
22             console.log(data.age);
23             if (data.age >= 1 && data.age <= 30) {
24                 ageRanges[0]++;
25             }
26             if (data.age >= 31 && data.age <= 40) {
27                 ageRanges[1]++;
28             }
29             if (data.age >= 41 && data.age <= 60) {
30                 ageRanges[2]++;
31             }
32             if (data.age >= 61 && data.age <= 100) {
33                 ageRanges[3]++;
34             }
35         }).then(function () {
36             var div = d3.select("body")
37                 .selectAll("div")
38                 .data(ageRanges)
39                 .enter()
40                 .append('div')
41                 .text(function (val, index) {
42                     returnText = '';
43                     switch(index) {
44                         case 0:
45                             returnText = "Ages 1 to 30: ";
46                             break;
47                         case 1:
48                             returnText = "Ages 31 to 40: ";
49                             break;
50                         case 2:
51                             returnText = "Ages 41 to 60: ";
52                             break;
53                         case 3:
54                             returnText = "Ages 61 to 100: ";
55                             break;
56                     }
57                     return returnText + val;
58                 })
59         })

```

ex11

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script src='https://d3js.org/d3.v7.min.js'></script>
6  <script>
7      //Create SVG element
8      var svg = d3.select("body")
9          .append("svg")
10         .attr("width", 400)
11         .attr("height", 400)
12         .style("border", '1px solid gold');
13
14     //Create line element inside SVG
15     svg.append("line")
16         .attr("x1", 100)
17         .attr("x2", 300)
18         .attr("y1", 100)
19         .attr("y2", 100)
20         .attr("stroke", "blue")
21
22     svg.append("line")
23         .attr("x1", 300)
24         .attr("x2", 300)
25         .attr("y1", 100)
26         .attr("y2", 300)
27         .attr("stroke", "red")
28
29     svg.append("line")
30         .attr("x1", 300)
31         .attr("x2", 100)
32         .attr("y1", 300)
33         .attr("y2", 300)
34         .attr("stroke", "green")
35
36     svg.append("line")
37         .attr("x1", 100)
38         .attr("x2", 100)
39         .attr("y1", 300)
40         .attr("y2", 100)
41         .attr("stroke", "yellow")
42
43 </script>
44
45 </body>
46 </html>
```

ex12

```
1   <!DOCTYPE html>
2   <html lang="en">
3
4   <body>
5     <script src='https://d3js.org/d3.v7.min.js'></script>
6     <script>
7
8       //Create SVG element
9       var svg = d3.select("body")
10      .append("svg")
11      .attr("width", 400)
12      .attr("height", 400)
13      .style("border", '1px solid green');
14
15     let shapescsv = 'https://raw.githubusercontent.com/HusainMehdi/DataVis/main/shapes.csv';
16
17   d3.csv(shapescsv, function (data) {
18     console.log(data.shape);
19     if (data.shape === "rect") {
20       console.log(data.x)
21       svg.append("rect")
22         .attr("x", data.x)
23         .attr("y", data.y)
24         .attr("width", data.width)
25         .attr("height", data.height)
26         .attr("fill", data.colour);
27     }
28     if (data.shape === "circle") {
29       console.log(data.x)
30       svg.append("circle")
31         .attr("cx", data.x)
32         .attr("cy", data.y)
33         .attr("r", data.r)
34         .attr("fill", data.colour);
35     }
36     if (data.shape === "ellipse") {
37       console.log(data.x)
38       svg.append("ellipse")
39         .attr("cx", data.x)
40         .attr("cy", data.y)
41         .attr("rx", data.rx)
42         .attr("ry", data.ry)
43         .attr("fill", data.colour);
44     }
45     if (data.shape === "line") {
46       console.log(data.x)
47       svg.append("line")
48         .attr("x1", data.x)
49         .attr("y1", data.y)
50         .attr("x2", data.x2)
51         .attr("y2", data.y2)
52         .attr("stroke", data.colour);
53     }
54   })
55
56
57   </script>
58
59 </body>
60
```

ex13

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

<body>
  <script src='https://d3js.org/d3.v7.min.js'></script>
  <script>

    //Create SVG element
    var svg = d3.select("body")
      .append("svg")
      .attr("width", 400)
      .attr("height", 400)
      .style("border", '1px solid green');

    let shapescsv = 'https://raw.githubusercontent.com/HusainMehdi/DataVis/main/shapes.csv';

    let shape0objs = [];

    d3.csv(shapescsv, function (data) {

      shape0objs.push({shape: data.shape,"x": data.x,"y": data.y,"x2": data.x2,"y2": data.y2,"r": data.r,"rx": data.rx,"ry": data.ry,"wid
    }).then(function () {

      console.log("hey");

      var enterShape = svg.selectAll("myShape")
        .data(shape0objs)
        .enter()
        .append(function(obj, i){
          console.log(obj.shape);
          switch (obj.shape) {
            case 'rect':
              return document.createElementNS('http://www.w3.org/2000/svg', 'rect');
              break;
            case 'circle':
              return document.createElementNS('http://www.w3.org/2000/svg', 'circle');
              break;
            case 'ellipse':
              return document.createElementNS('http://www.w3.org/2000/svg', 'ellipse');
              break;
            case 'line':
              return document.createElementNS('http://www.w3.org/2000/svg', 'line');
              break;
          }
        })
        .attr("x", function(obj){return obj.x})
        .attr("y", function(obj){return obj.y})
        .attr("x1", function(obj){return obj.x})
        .attr("y1", function(obj){return obj.y})
        .attr("x2", function(obj){return obj.x2})
        .attr("y2", function(obj){return obj.y2})
        .attr("cx", function(obj){return obj.x})
        .attr("cy", function(obj){return obj.y})
        .attr("r", function(obj){return obj.r})
        .attr("rx", function(obj){if(obj.shape === 'rect'){return 0} else{return obj.rx}})
        .attr("ry", function(obj){if(obj.shape === 'rect'){return 0} else{return obj.ry}})

    });

  
```

ex14

```

<style>
    svg rect {
        fill: blue;
    }

    svg text {
        fill: white;
        font: 10px sans-serif;
        text-anchor: end;
    }
</style>

<script>
    let heartfailurecsv = 'https://raw.githubusercontent.com/akmand/datasets/master/heart\_failure.csv';

    var ageRanges = [0, 0, 0, 0]

    d3.csv(heartfailurecsv, function (data) {
        console.log(data.age);
        if (data.age >= 1 && data.age <= 30) {
            ageRanges[0]++;
        }
        if (data.age >= 31 && data.age <= 40) {
            ageRanges[1]++;
        }
        if (data.age >= 41 && data.age <= 60) {
            ageRanges[2]++;
        }
        if (data.age >= 61 && data.age <= 100) {
            ageRanges[3]++;
        }
    }).then(function () {
        var width = 200;
        var scaleFactor = 5;
        var barHeight = 20;

        var graph = d3.select("body")
            .append("svg")
            .attr("width", 1000)
            .attr("height", barHeight * ageRanges.length);
        var bar = graph.selectAll("g")
            .data(ageRanges)
            .enter()
            .append("g")
            .attr("transform", function (d, i) {
                return "translate(0," + i * barHeight + ")";
            });
        bar.append("rect")
            .attr("width", function (d) {
                return d * scaleFactor;
            })
            .attr("height", barHeight - 1);
        bar.append("text")
            .attr("x", function (d) { if (d==0){return 10} else {return ((d * scaleFactor)-2); }})
            .attr("y", barHeight / 2)
            .attr("dy", ".35em")
            .text(function (d) { return d; })
            .style("fill", "pink");
    });
</script>

```

ex15

```

<style>
  svg text {
    fill: white;
    font: 10px sans-serif;
    text-anchor: end;
  }
</style>

<script>

let heartfailurecsv = 'https://raw.githubusercontent.com/akmand/datasets/master/heart_failure.csv';

var ageRanges = [0, 0, 0, 0]

d3.csv(heartfailurecsv, function (data) {
  console.log(data.age);
  if (data.age >= 1 && data.age <= 30) {
    ageRanges[0]++;
  }
  if (data.age >= 31 && data.age <= 40) {
    ageRanges[1]++;
  }
  if (data.age >= 41 && data.age <= 60) {
    ageRanges[2]++;
  }
  if (data.age >= 61 && data.age <= 100) {
    ageRanges[3]++;
  }
}).then(function () {
  var width = 200;
  var scaleFactor = 5;
  var barHeight = 20;

  var graph = d3.select("body")
    .append("svg")
    .attr("width", 1000)
    .attr("height", barHeight * ageRanges.length);
  var bar = graph.selectAll("g")
    .data(ageRanges)
    .enter()
    .append("g")
    .attr("transform", function (d, i) {
      return "translate(0," + i * barHeight + ")";
    });
  bar.append("rect")
    .attr("width", function (d) {
      return d * scaleFactor;
    })
    .attr("height", barHeight - 1)
    .attr("fill", function (d) { if (d > 100) { return "darkred" } else { return "darkgreen" } })
  bar.append("text")
    .attr("x", function (d) { if (d === 0) { return 10 } else { return ((d * scaleFactor) - 2); } })
    .attr("y", barHeight / 2)
    .attr("dy", ".35em")
    .text(function (d) { return d; })
    .style("fill", function (d) { if (d > 100) { return "pink" } else { return "darkseagreen" } });
}

```

ex16

```
<script>
  const width = 500;
  const height = 500;
  const data = [10, 15, 20, 25, 30];
  // Note different valid ways of specifying color
  const colors = ['#ffffcc', 'red', 'rgb(0,255,0)', '#31a354', '#006837'];
  const svg = d3.select("body")
    .append("svg")
    .attr("width", width)
    .attr("height", height);

  const g = svg.selectAll("g")
    .data(data)
    .enter()
    .append("g")
    .attr("transform", function (d, i) {
      return "translate(0,0)";
    })

  g.append(function (d, i) {
    switch (data[i] % 10) {
      case 0:
        return document.createElementNS('http://www.w3.org/2000/svg', 'rect');
        break;
      case 5:
        return document.createElementNS('http://www.w3.org/2000/svg', 'circle');
        break;
    }
  })
    .attr("cx", function (d, i) {
      return i * 100 + 50;
    })
    .attr("cy", function (d, i) {
      return 100;
    })
    .attr("r", function (d) {
      return d * 1.5;
    })
    .attr("fill", function (d, i) {
      return colors[i];
    })
    .attr("x", function (d, i) { return (i * 100 + 50) - (d * 1.5) })
    .attr("y", function (d, i) { return 100 - (d * 1.5) })
    .attr("width", function (d, i) { return d * 3 })
    .attr("height", function (d, i) { return d * 3 })

  g.append("text")
    .attr("x", function (d, i) {
      return i * 100 + 40;
    })
    .attr("y", 105)
    .attr("stroke", "teal")
    .attr("font-size", "12px")
    .attr("font-family", "sans-serif")
    .text(function (d) {
      return d;
    });
</script>
```

ex17

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script src='https://d3js.org/d3.v7.min.js'></script>
6
7      <script>
8          const data = [50, 400, 300, 900, 250, 1000]
9          const width = 500;
10         const barHeight = 20;
11         const margin = 1;
12         var scale = d3.scaleLinear()
13             .domain([d3.min(data), d3.max(data)])
14             .range([50, 500]);
15
16         var svg = d3.select("body")
17             .append("svg")
18             .attr("width", width)
19             .attr("height", barHeight * data.length);
20
21         var g = svg.selectAll("g")
22             .data(data)
23             .enter()
24             .append("g")
25             .attr("transform", function (d, i) {
26                 return "translate(0," + i * barHeight + ")";
27             });
28
29         g.append("rect")
30             .attr("width", function (d) {
31                 return scale(d);
32             })
33             .attr('fill', function (d) { if (d > 500) { return "red" } else { return "green" } })
34             .attr("height", barHeight - margin)
35
36         g.append("text")
37             .attr("x", function (d) { return (scale(d)); })
38             .attr("y", barHeight / 2)
39             .attr("dy", ".35em")
40             .style('text-anchor', 'end')
41             .text(function (d) { return d; });
42     </script>
43
44 </body>
45
46 </html>
```

ex18

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script src='https://d3js.org/d3.v7.min.js'></script>
6
7      <script>
8          const data = [50, 400, 300, 900, 250, 1000]
9          const width = 500;
10         const barscsv: string
11         const barsarr: Array<number> = [];
12         let barscsv = 'https://raw.githubusercontent.com/HusainMehdi/DataVis/main/bars.csv';
13         var barsarr = new Array<number>();
14         let barscsv = 'https://raw.githubusercontent.com/HusainMehdi/DataVis/main/bars.csv';
15         var barsarr = new Array<number>();
16
17         d3.csv(barscsv, function (data) {
18
19             barsarr.push(parseInt(data.val));
20
21         })
22         .then(function () {
23
24             var scale = d3.scaleLinear()
25                 .domain([d3.min(barsarr), d3.max(barsarr)])
26                 .range([50, 500]);
27
28             var svg = d3.select("body")
29                 .append("svg")
30                 .attr("width", width)
31                 .attr("height", barHeight * barsarr.length + 1);
32             console.log(barsarr);
33             var g = svg.selectAll("g")
34                 .data(barsarr)
35                 .enter()
36                 .append("g")
37                 .attr("transform", function (d, i) {
38                     return "translate(0," + i * barHeight + ")";
39                 });
40
41             g.append("rect")
42                 .attr("width", function (d) {
43                     return scale(d);
44                 })
45                 .attr("fill", function (d) { if (d > 500) { return "red" } else { return "green" } })
46                 .attr("height", barHeight - margin);
47
48             g.append("text")
49                 .attr("x", function (d) { return (scale(d)); })
50                 .attr("y", barHeight / 2)
51                 .attr("dy", ".35em")
52                 .style('text-anchor', 'end')
53                 .text(function (d) { return d; });
54
55         })
56
57     </script>
58
59 
```

ex19

```

1   <!DOCTYPE html>
2   <html lang="en">
3
4   <body>
5       <script src='https://d3js.org/d3.v7.min.js'></script>
6
7   <script>
8
9   function createBar(inputcsv) {
10      const data = [50, 400, 300, 900, 250, 1000]
11      const width = 500;
12      const barHeight = 20;
13      const margin = 1;
14
15      let barscsv = inputcsv;
16      var barsarr = new Array();
17
18      d3.csv(barscsv, function (data) {
19
20          barsarr.push(parseInt(data.val));
21
22      })
23      .then(function () {
24
25          var scale = d3.scaleLinear()
26              .domain([d3.min(barsarr), d3.max(barsarr)])
27              .range([50, 500]);
28
29          var svg = d3.select("body")
30              .append("svg")
31              .attr("width", width)
32              .attr("height", barHeight * barsarr.length + 1);
33          console.log(barsarr);
34          var g = svg.selectAll("g")
35              .data(barsarr)
36              .enter()
37              .append("g")
38              .attr("transform", function (d, i) {
39                  return "translate(0," + i * barHeight + ")";
40              });
41
42          g.append("rect")
43              .attr("width", function (d) {
44                  return scale(d);
45              })
46              .attr('fill', function (d) { if (d > 500) { return "red" } else { return "green" } })
47              .attr("height", barHeight - margin)
48
49          g.append("text")
50              .attr("x", function (d) { return (scale(d)); })
51              .attr("y", barHeight / 2)
52              .attr("dy", ".35em")
53              .style('text-anchor', 'end')
54              .text(function (d) { return d; });
55      })
56
57      createBar('https://raw.githubusercontent.com/HusainMehdi/DataVis/main/bars.csv');
58      createBar('https://raw.githubusercontent.com/HusainMehdi/DataVis/main/bars2.csv');
59
60  </script>

```

ex20

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7          var svg: any
8          var svg = d3.select("body")
9          .append("svg")
10         var svg = d3.select("body")
11         .append("svg")
12         .attr("width", width)
13         .attr("height", height);
14         var xscale = d3.scaleLinear()
15         .domain([0, d3.max(data)])
16         .range([0, width - 100]);
17         var yscale = d3.scaleLinear()
18         .domain([0, d3.max(data)])
19         .range([height / 2, 0]);
20         var x_axis = d3.axisBottom()
21         .scale(xscale);
22         var y_axis = d3.axisLeft()
23         .scale(yscale);
24         var x_axis2 = d3.axisTop()
25         .scale(xscale);
26         var y_axis2 = d3.axisRight()
27         .scale(yscale);
28         svg.append("g")
29         .attr("transform", "translate(50, 20)")
30         .call(y_axis);
31         var xAxisTranslate = height / 2 + 20;
32         svg.append("g")
33         .attr("transform", "translate(50, " + xAxisTranslate + ")")
34         .call(x_axis);
35         svg.append("g")
36         .attr("transform", "translate(50, 20)");
37         .attr("stroke", "blue")
38         .attr("color", "blue")
39         .call(x_axis2);
40         var yAxisTranslate = width - 50;
41         svg.append("g")
42         .attr("transform", "translate(" + yAxisTranslate + ", 20)")
43         .attr("stroke", "blue")
44         .attr("color", "blue")
45         .call(y_axis2)
46     </script>
47
48 </body>
49
50 </html>
```

ex21

```
8      const data = [30, 400, 300, 300, 250, 1000]
9      const width = 500;
10     const barHeight = 20;
11     const margin = 1;
12     const height = 100;
13     var scale = d3.scaleLinear()
14         .domain([d3.min(data), d3.max(data)])
15         .range([50, 500]);
16
17     var svg = d3.select("body")
18         .append("svg")
19         .attr("width", width)
20         .attr("height", barHeight * data.length + 20);
21
22     var g = svg.selectAll("g")
23         .data(data)
24         .enter()
25         .append("g")
26         .attr("transform", function (d, i) {
27             return "translate(0," + i * barHeight + ")";
28         });
29
30     var xscale = d3.scaleLinear()
31         .domain([0, d3.max(data)])
32         .range([0, width]);
33
34     var x_axis = d3.axisBottom()
35         .scale(xscale);
36
37     var yscale = d3.scaleLinear()
38         .domain([5, 0])
39         .range([barHeight * data.length, 0]);
40
41     var y_axis = d3.axisRight()
42         .scale(yscale)
43         .ticks(data.length);
44
45     g.append("rect")
46         .attr("width", function (d) {
47             return scale(d);
48         })
49         .attr('fill', function (d) { if (d > 500) { return "red" } else { return "green" } })
50         .attr("height", barHeight - margin)
51
52     g.append("text")
53         .attr("x", function (d) { return (scale(d)); })
54         .attr("y", barHeight / 2)
55         .attr("dy", ".35em")
56         .style('text-anchor', 'end')
57         .text(function (d) { return d; });
58
59     var xAxisTranslate = barHeight * data.length;
60     svg.append("g")
61         .attr("transform", "translate(0, " + xAxisTranslate + ")")
62         .call(x_axis)
63
64     svg.append("g")
65         .attr("transform", "translate(0, 0)")
66         .call(y_axis);
```

~/Repo
1 prob

ex22

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <body>
5      <script src='https://d3js.org/d3.v7.min.js'></script>
6      <script>
7          function drawLine(wave) {
8              // Set Dimensions
9              const xSize = 600; const ySize = 600;
10             const margin = 40;
11             const xMax = xSize - margin * 2;
12             const yMax = ySize - margin * 2;
13
14             // Create Random Points
15             const numPoints = 100;
16             const data = [];
17             switch (wave) {
18                 case 'sin':
19                     for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.sin(6.2 * i / 100) }); }
20                     break;
21                 case 'cos':
22                     for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.cos(6.2 * i / 100) }); }
23                     break;
24                 case 'tan':
25                     for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.tan(6.2 * i / 100) }); }
26                     break;
27             }
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(data, d => { return d.x });
33             const yExtent = d3.extent(data, 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));
57
58             // Y Axis
```

```
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      // Add the line
73      svg.append("path")
74          .datum(data)
75          .attr("fill", "none")
76          .attr("stroke", "steelblue")
77          .attr("stroke-width", 1.5)
78          .attr("d", d3.line()
79              .x(function (d) { return x(d.x) })
80              .y(function (d) { return y(d.y) })
81          );
82      }
83
84      drawLine('cos');
85  </script>
86
87  </body>
88
89  </html>
```

ex23

```

4  <body>
5    <script src='https://d3js.org/d3.v7.min.js'></script>
6    <script>
7      function drawLine(wave) {
8        // Set Dimensions
9        const xSize = 600; const ySize = 600;
10       const margin = 40;
11       const xMax = xSize - margin * 2;
12       const yMax = ySize - margin * 2;
13
14
15      let sincsv = 'https://raw.githubusercontent.com/HusainMehdi/DataVis/main/sin.csv';
16      let sinarray = [];
17
18
19      d3.csv(sincsv, function (data) {
20        sinarray.push({ x: parseFloat(data.x), y: parseFloat(data.y) });
21      })
22        .then(function () {
23          console.log(sinarray)
24
25          // Create Random Points
26          const numPoints = 100;
27          data = [];
28          switch (wave) {
29            case 'sin':
30              data = sinarray;
31              break;
32            case 'cos':
33              for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.cos(6.2 * i / 100) }); }
34              break;
35            case 'tan':
36              for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.tan(6.2 * i / 100) }); }
37              break;
38          }
39
40
41          // Get the 'limits' of the data - the full extent (mins and max)
42          // so the plotted data fits perfectly
43          const xExtent = d3.extent(data, d => { return d.x });
44          const yExtent = d3.extent(data, d => { return d.y });
45
46          // Append SVG Object to the Page
47          const svg = d3.select("body")
48            .append("svg")
49              .attr('width', xSize)
50              .attr('height', ySize)
51              .append("g")
52                .attr("transform", "translate(" + margin + "," + margin + ")");
53
54          // X Axis
55          const x = d3.scaleLinear()
56            .domain([xExtent[0], xExtent[1]])
57            .range([0, xMax]);

```

ex

```

39
40
41         // Get the 'limits' of the data - the full extent (mins and max)
42         // so the plotted data fits perfectly
43         const xExtent = d3.extent(data, d => { return d.x });
44         const yExtent = d3.extent(data, d => { return d.y });
45
46         // Append SVG Object to the Page
47         const svg = d3.select("body")
48             .append("svg")
49             .attr('width', xSize)
50             .attr('height', ySize)
51             .append("g")
52             .attr("transform", "translate(" + margin + "," + margin + ")");
53
54         // X Axis
55         const x = d3.scaleLinear()
56             .domain([xExtent[0], xExtent[1]])
57             .range([0, xMax]);
58
59         // bottom
60         svg.append("g")
61             .attr("transform", "translate(0," + yMax + ")")
62             .call(d3.axisBottom(x))
63             .attr('color', 'green'); // make bottom axis green
64
65         // top
66         svg.append("g")
67             .call(d3.axisTop(x));
68
69         // Y Axis
70         const y = d3.scaleLinear()
71             .domain([yExtent[0], yExtent[1]])
72             .range([yMax, 0]);
73
74         // left y axis
75         svg.append("g")
76             .call(d3.axisLeft(y));
77
78         // right y axis
79         svg.append("g")
80             .attr("transform", `translate(${yMax},0)`)
81             .call(d3.axisRight(y));
82
83         // Add the line
84         svg.append("path")
85             .datum(data)
86             .attr("fill", "none")
87             .attr("stroke", "steelblue")
88             .attr("stroke-width", 1.5)
89             .attr("any", d3.line()
90                 .x(function (d) { return x(d.x) })
91                 .y(function (d) { return y(d.y) })
92             );
93
94     }
95
96     drawLine('sin');

```

```

1   <!DOCTYPE html>
2   <html lang="en">
3
4   <body>
5     <script src='https://d3js.org/d3.v7.min.js'></script>
6     <script>
7       // Set Dimensions
8       const xSize = 600; const ySize = 600;
9       const margin = 40;
10      const xMax = xSize - margin * 2;
11      const yMax = ySize - margin * 2;
12
13      // Create Random Points
14      const numPoints = 100;
15      const data = [];
16      for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.sin(6.2 * i / 100) }); }
17
18      const data2 = [];
19      for (let i = 0; i < numPoints; i++) { data2.push({ x: i / 100, y: Math.cos(6.2 * i / 100) }); }
20
21
22     // Get the 'limits' of the data – the full extent (mins and max)
23     // so the plotted data fits perfectly
24     const xExtent = d3.extent(data, d => { return d.x });
25     const yExtent = d3.extent(data, d => { return d.y });
26
27     // Append SVG Object to the Page
28     const svg = d3.select("body")
29       .append("svg")
30       .attr('width', xSize)
31       .attr('height', ySize)
32       .append("g")
33       .attr("transform", "translate(" + margin + "," + margin + ")");
34
35     // X Axis
36     const x = d3.scaleLinear()
37       .domain([xExtent[0], xExtent[1]])
38       .range([0, xMax]);
39
40     // bottom
41     svg.append("g")
42       .attr("transform", "translate(0," + yMax + ")")
43       .call(d3.axisBottom(x))
44       .attr('color', 'green'); // make bottom axis green
45
46     // top
47     svg.append("g")
48       .call(d3.axisTop(x));
49
50     // Y Axis
51     const y = d3.scaleLinear()
52       .domain([yExtent[0], yExtent[1]])
53       .range([yMax, 0]);
54
55     // left y axis
56     svg.append("g")
57       .call(d3.axisLeft(y));
58

```

```
54
55     // left y axis
56     svg.append("g")
57         .call(d3.axisLeft(y));
58
59     // right y axis
60     svg.append("g")
61         .attr("transform", `translate(${yMax},0`)
62         .call(d3.axisRight(y));
63
64     // Add the line
65     svg.append("path")
66         .datum(data)
67         .attr("fill", "none")
68         .attr("stroke", "steelblue")
69         .attr("stroke-width", 1.5)
70     ▼       .attr("d", d3.line()
71             |   .x(function (d) { return x(d.x) })
72             |   .y(function (d) { return y(d.y) })
73     );
74
75     // Add the line
76     svg.append("path")
77         .datum(data2)
78             |   .attr("fill", "none")
79             |   .attr("stroke", "green")
80             |   .attr("stroke-width", 1.5)
81     ▼       .attr("d", d3.line()
82             |   .x(function (d) { return x(d.x) })
83             |   .y(function (d) { return y(d.y) })
84     );
85     </script>
86
87 </body>
88
89 </html>
```

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

```
56
57     // right y axis
58     svg.append("g")
59         .attr("transform", `translate(${yMax},0)`)
60         .call(d3.axisRight(y));
61
62     // Add the line
63     svg.append("path")
64         .datum(data)
65         .attr("fill", "none")
66         .attr("stroke", "steelblue")
67         .attr("stroke-width", 1.5)
68         .attr("d", d3.line()
69             .x(function (d) { return x(d.x) })
70             .y(function (d) { return y(d.y) })
71         );
72
73
74     svg.selectAll("dot")
75         .data(data)
76         .enter()
77         .append("circle")
78         .attr("cx", function (d) { return x(d.x) })
79         .attr("cy", function (d) { return y(d.y) })
80         .attr("r", 5)
81         .style("fill", "red");
82
83
84     </script>
85
86 </body>
87
88 </html>
```

```

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

```

```

58      // right y axis
59      svg.append("g")
60          .attr("transform", `translate(${yMax},0`)
61          .call(d3.axisRight(y));
62
63      // Add the line
64      svg.append("path")
65          .datum(data)
66          .attr("fill", "none")
67          .attr("stroke", "steelblue")
68          .attr("stroke-width", 1.5)
69          .attr("d", d3.line()
70              .x(function (d) { return x(d.x) })
71              .y(function (d) { return y(d.y) })
72          );
73
74      // Add the line
75      svg.append("path")
76          .datum(data2)
77          .attr("fill", "none")
78          .attr("stroke", "green")
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      svg.selectAll("dot")
86          .data(data)
87          .enter()
88          .append("circle")
89          .attr("cx", function (d) { return x(d.x) })
90          .attr("cy", function (d) { return y(d.y) })
91          .attr("r", 5)
92          .style("fill", "red");
93
94      var triangle = d3.symbol()
95      .....
96      .....
97      .....
98
99
100     svg.selectAll("dot")
101         .data(data2)
102         .enter()
103         .append("path")
104         .attr("d", triangle)
105         .attr("transform", function (d) { return "translate("+x(d.x)+","+y(d.y)+"") })
106         .style("fill", "purple");
107
108     </script>
109
110 </body>
111
112 </html>

```

```

3
4 <body>
5   <script src='https://d3js.org/d3.v7.min.js'></script>
6   <script>
7     // Set Dimensions
8     const xSize = 600; const ySize = 600;
9     const margin = 40;
10    const xMax = xSize - margin * 2;
11    const yMax = ySize - margin * 2;
12
13    // Create Random Points
14    const numPoints = 100;
15    const data = [];
16    for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.sin(6.2 * i / 100) }); }
17
18    const data2 = [];
19    for (let i = 0; i < numPoints; i++) { data2.push({ x: i / 100, y: Math.cos(6.2 * i / 100) }); }
20
21
22    // Get the 'limits' of the data - the full extent (mins and max)
23    // so the plotted data fits perfectly
24    const xExtent = d3.extent(data, d => { return d.x });
25    const yExtent = d3.extent(data, d => { return d.y });
26
27    // Append SVG Object to the Page
28    const svg = d3.select("body")
29      .append("svg")
30        .attr('width', xSize)
31        .attr('height', ySize)
32        .append("g")
33        .attr("transform", "translate(" + margin + "," + margin + ")");
34
35    // X Axis
36    const x = d3.scaleLinear()
37      .domain([xExtent[0], xExtent[1]])
38      .range([0, xMax]);
39
40    // bottom
41    svg.append("g")
42      .attr("transform", "translate(0," + yMax + ")")
43      .call(d3.axisBottom(x))
44      .attr('color', 'green'); // make bottom axis green
45
46    // top
47    svg.append("g")
48      .call(d3.axisTop(x));
49
50    // Y Axis
51    const y = d3.scaleLinear()
52      .domain([yExtent[0], yExtent[1]])
53      .range([yMax, 0]);
54
55    // left y axis
56    svg.append("g")
57      .call(d3.axisLeft(y));
58

```

```

59     // right y axis
60     svg.append("g")
61         .attr("transform", `translate(${yMax},0`)
62         .call(d3.axisRight(y));
63
64     // Add the line
65     svg.append("path")
66         .datum(data)
67         .attr("fill", "none")
68         .attr("stroke", "steelblue")
69         .attr("stroke-width", 1.5)
70         .attr("d", d3.line()
71             .x(function (d) { return x(d.x) })
72             .y(function (d) { return y(d.y) })
73         );
74
75     // Add the line
76     svg.append("path")
77         .datum(data2)
78         .attr("fill", "none")
79         .attr("stroke", "green")
80         .attr("stroke-width", 1.5)
81         .attr("d", d3.line()
82             .x(function (d) { return x(d.x) })
83             .y(function (d) { return y(d.y) })
84         );
85
86     svg.selectAll("dot")
87         .data(data)
88         .enter()
89         .append("circle")
90         .attr("cx", function (d) { return x(d.x) })
91         .attr("cy", function (d) { return y(d.y) })
92         .attr("r", 5)
93         .style("fill", "red")
94         .text("hi");
95
96     var triangle = d3.symbol()
97         .type(d3.symbolTriangle)
98         .size(50)
99
100
101    svg.selectAll("dot")
102        .data(data2)
103        .enter()
104        .append("path")
105        .attr("d", triangle)
106        .attr("transform", function (d) { return "translate(" + x(d.x) + "," + y(d.y) + ")" })
107        .style("fill", "purple");
108
109    svg.selectAll("dot")
110        .data(data)
111        .enter()
112        .append("text")
113        .attr("x", function(d){return x(d.x)})
114        .attr("y", function(d){return y(d.y)})
115        .attr("dy", ".35em")
116        .style('text-anchor', 'end')
117        .text(function(d){return parseInt(y(d.y))})
118        .attr("opacity", function(d,i){if(i % 10 === 0){ return 1} else {return 0}});

```

```
15 <script>
16     var body = d3.select('body');
17     var myColor0 = d3.scaleLinear().domain([1, 10]).range(["white", "blue"]);
18     var myColor1 = d3.scaleSequential().domain([1, 10]).interpolator(d3.interpolateOrRd);
19     var myColor2 = d3.scaleSequential().domain([1, 10]).interpolator(d3.interpolateViridis);
20 /*
21     d3.range(5)
22     [0, 1, 2, 3, 4]
23 */
24     var data2 = d3.range(5);
25     var myColor3 = d3.scaleOrdinal().domain(data2).range(["gold", "blue", "green", "yellow"]);
26     var myColor4 = d3.scaleOrdinal().domain(data2).range(d3.schemeSet3);
27     var cols = [myColor0,
28         myColor1,
29         myColor2,
30         myColor3,
31         myColor4];
32
33 const data = [50, 400, 300, 900, 250, 1000]
34 const width = 500;
35 const barHeight = 20;
36 const margin = 1;
37 var scale = d3.scaleLinear()
38     .domain([d3.min(data), d3.max(data)])
39     .range([50, 500]);
40
41 var scale2 = d3.scaleLinear()
42     .domain([d3.min(data), d3.max(data)])
43     .range([1, 10]);
44
45 var svg = d3.select("body")
46     .append("svg")
47     .attr("width", width)
48     .attr("height", barHeight * data.length);
49
50 var g = svg.selectAll("g")
51     .data(data)
52     .enter()
53     .append("g")
54     .attr("transform", function (d, i) {
55         return "translate(0," + i * barHeight + ")";
56     });
57
58 g.append("rect")
59     .attr("width", function (d) {
60         return scale(d);
61     })
62     .attr('fill', function (d) {
63         return myColor1(scale2(d));
64     })
65     .attr("height", barHeight - margin)
66
67 g.append("text")
68     .attr("x", function (d) { return (scale(d)); })
69     .attr("y", barHeight / 2)
70     .attr("dy", ".35em")
71     .style('text-anchor', 'end')
72     .text(function (d) { return d; });
```

```
<script>

    var body = d3.select('body');
    var myColor0 = d3.scaleLinear().domain([1, 10]).range(["white", "blue"]);
    var myColor1 = d3.scaleSequential().domain([1, 10]).interpolator(d3.interpolateCool);
    var myColor2 = d3.scaleSequential().domain([1, 10]).interpolator(d3.interpolateViridis);
    /*
    d3.range(5)
    [0, 1, 2, 3, 4]
    */
    var data2 = d3.range(5);
    var myColor3 = d3.scaleOrdinal().domain(data2).range(["gold", "blue", "green", "yellow"]);
    var myColor4 = d3.scaleOrdinal().domain(data2).range(d3.schemeSet3);
    var cols = [myColor0,
        myColor1,
        myColor2,
        myColor3,
        myColor4];

    // Set Dimensions
    const xSize = 600; const ySize = 600;
    const margin = 40;
    const xMax = xSize - margin * 2;
    const yMax = ySize - margin * 2;

    // Create Random Points
    const numPoints = 100;
    const data = [];
    for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.sin(6.2 * i / 100) }); }

    // Get the 'limits' of the data - the full extent (mins and max)
    // so the plotted data fits perfectly
    const xExtent = d3.extent(data, d => { return d.x });
    const yExtent = d3.extent(data, d => { return d.y });

    // Append SVG Object to the Page
    const svg = d3.select("body")
        .append("svg")
        .attr('width', xSize)
        .attr('height', ySize)
        .append("g")
        .attr("transform", "translate(" + margin + "," + margin + ")");


```

```

// X Axis
const x = d3.scaleLinear()
    .domain([xExtent[0], xExtent[1]])
    .range([0, xMax]);

// bottom
svg.append("g")
    .attr("transform", "translate(0," + yMax + ")")
    .call(d3.axisBottom(x))
    .attr('color', 'green'); // make bottom axis green

// top
svg.append("g")
    .call(d3.axisTop(x));

// Y Axis
const y = d3.scaleLinear()
    .domain([yExtent[0], yExtent[1]])
    .range([yMax, 0]);

// left y axis
svg.append("g")
    .call(d3.axisLeft(y));

// right y axis
svg.append("g")
    .attr("transform", `translate(${yMax},0)`)
    .call(d3.axisRight(y));

// Add the line
svg.append("path")
    .datum(data)
    .attr("fill", "none")
    .attr("stroke", "steelblue")
    .attr("stroke-width", 1.5)
    .attr("d", d3.line()
        .x(function (d) { return x(d.x) })
        .y(function (d) { return y(d.y) })
    );

// dots
svg.selectAll("dot")
    .data(data)
    .enter()
    .append("circle")
    .attr("cx", function (d) { return x(d.x) })
    .attr("cy", function (d) { return y(d.y) })
    .attr("r", 5)
    .style("fill", function (d) {
        return myColor1(d.x*10);
    });

```

```

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 data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12];
9       const xSize = 400; const ySize = 400;
10      const margin = 40;
11      const xMax = xSize - margin * 2;
12      const yMax = ySize - margin * 2;
13
14      // Append SVG Object to the Page
15      const svg = d3.select("body")
16          .append("svg")
17          .attr('width', xSize)
18          .attr('height', ySize)
19          .append("g")
20          .attr("transform", "translate(" + xSize / 2 + "," + ySize / 2 + ")");
21      const radius = Math.min(xSize, ySize) / 2;
22      var color = d3.scaleOrdinal(['#4daf4a', '#377eb8', '#ff7f00', '#984ea3', '#e41a1c']);
23
24      // Generate the pie
25      var pie = d3.pie()
26          .padAngle(0);
27
28      // Generate the arcs
29      var arc = d3.arc()
30          .innerRadius(0)
31          .outerRadius(radius);
32
33      //Generate groups
34      var arcs = svg.selectAll("arc")
35          .data(pie(data))
36          .enter()
37          .append("g")
38          .attr("class", "arc");
39
40      //Draw arc paths
41      arcs.append("path")
42          .attr("fill", function (d, i) {
43              return color(i);
44          })
45          .attr("d", arc);
46      </script>
47
48  </body>
49
50  </html>
```

```

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 data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12];
9      const xSize = 400; const ySize = 400;
10     const margin = 40;
11     const xMax = xSize - margin * 2;
12     const yMax = ySize - margin * 2;
13
14     // Append SVG Object to the Page
15     const svg = d3.select("body")
16         .append("svg")
17         .attr('width', xSize)
18         .attr('height', ySize)
19         .append("g")
20         .attr("transform", "translate(" + xSize / 2 + "," + ySize / 2 + ")");
21     const radius = Math.min(xSize, ySize) / 2;
22     var color = d3.scaleOrdinal(['#4daf4a', '#377eb8', '#ff7f00', '#984ea3', '#e41a1c']);
23
24     // Generate the pie
25     var pie = d3.pie()
26         .padAngle(0);
27
28     // Generate the arcs
29     var arc = d3.arc()
30         .innerRadius(0)
31         .outerRadius(radius);
32
33
34
35     //Generate groups
36     var arcs = svg.selectAll("arc")
37         .data(pie(data))
38         .enter()
39         .append("g")
40         .attr("class", "arc")
41
42     //Draw arc paths
43     arcs.append("path")
44         .attr("fill", function (d, i) {
45             return color(i);
46         })
47         .attr("d", arc);
48
49     const arcLabel = function () {
50         return d3.arc().innerRadius(100).outerRadius(200);
51     };
52
53     var text = arcs.append("text")
54         .text(function (d) { console.log(d); return d.data })
55         .attr("transform", (d) => `translate(${arcLabel().centroid(d)})`);
56
57
58 </script>
</body>
```

```
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 body = d3.select('body');
9     var myColor0 = d3.scaleLinear().domain([1, 10]).range(["white", "blue"]);
10    var myColor1 = d3.scaleSequential().domain([1, 10]).interpolator(d3.interpolateCool);
11    var myColor2 = d3.scaleSequential().domain([1, 10]).interpolator(d3.interpolateViridis);
12  /*
13    d3.range(5)
14    [0, 1, 2, 3, 4]
15  */
16  var data2 = d3.range(5);
17  var myColor3 = d3.scaleOrdinal().domain(data2).range(["gold", "blue", "green", "yellow"]);
18  var myColor4 = d3.scaleOrdinal().domain(data2).range(d3.schemeSet3);
19  var cols = [myColor0,
20    myColor1,
21    myColor2,
22    myColor3,
23    myColor4];
24
25
26
27 // Set Dimensions
28 const xSize = 600; const ySize = 600;
29 const margin = 40;
30 const xMax = xSize - margin * 2;
31 const yMax = ySize - margin * 2;
32
33 // Create Random Points
34 const numPoints = 100;
35 const data = [];
36 for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.sin(6.2 * i / 100) }); }
37
38
39
40
41 // Get the 'limits' of the data - the full extent (mins and max)
42 // so the plotted data fits perfectly
43 const xExtent = d3.extent(data, d => { return d.x });
44 const yExtent = d3.extent(data, d => { return d.y });
45
46 // Append SVG Object to the Page
47 const svg = d3.select("body")
48   .append("svg")
49   .attr('width', xSize)
50   .attr('height', ySize)
51   .append("g")
52   .attr("transform", "translate(" + margin + "," + margin + ")");
```

```

svg.append("svg:image")
    .attr("xlink:href", "https://raw.githubusercontent.com/HusainMehdi/DataVis/main/img.jpeg")
    .attr("width", xSize)
    .attr("height", ySize)
    .attr("x", 0)
    .attr("y", -margin)

// X Axis
const x = d3.scaleLinear()
    .domain([xExtent[0], xExtent[1]])
    .range([0, xMax]);

// bottom
svg.append("g")
    .attr("transform", "translate(0," + yMax + ")")
    .call(d3.axisBottom(x))
    .attr('color', 'green'); // make bottom axis green

// top
svg.append("g")
    .call(d3.axisTop(x));

// Y Axis
const y = d3.scaleLinear()
    .domain([yExtent[0], yExtent[1]])
    .range([yMax, 0]);

// left y axis
svg.append("g")
    .call(d3.axisLeft(y));

// right y axis
svg.append("g")
    .attr("transform", `translate(${yMax},0)`)
    .call(d3.axisRight(y));

// Add the line
svg.append("path")
    .datum(data)
    .attr("fill", "none")
    .attr("stroke", "steelblue")
    .attr("stroke-width", 1.5)
    .attr("d", d3.line()
        .x(function (d) { return x(d.x) })
        .y(function (d) { return y(d.y) }))
    );

// Add the dots
svg.selectAll("dot")
    .data(data)
    .enter()
    .append("circle")
    .attr("cx", function (d) { return x(d.x) })
    .attr("cy", function (d) { return y(d.y) })
    .attr("r", 5)
    .style("fill", function (d) {
        return myColor1(d.x * 10);
    });

```