3.1.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8"/>

    <meta name="description" content="Data Visualisation"/>

    <meta name="keyword" content="HTML, CSS"/>

    <meta name="author" content="Minh Nguyen"/>

    <title>Task 3.1 D3 Scale</title>

    <script src="https://d3js.org/d3.v5.min.js"></script>

    <script src="3.1.js"></script>

</head>

<body>

    <h1>Drawing with Data - Scatter Plot</h1>

    <div id="scatter-pot"></div>

    <br>

    <bf></bf>

    <footer style="color:grey">COS30045 Data Visualisation<br>Minh Nguyen</footer>

</body>

</html>

3.1.js

function init()

{

    var w = 600; // subtracting 40px for padding

    var h = 200; // subtracting 20px for padding

    var padding = 40;

    var dataset = [

                    [5, 20],

                    [500, 90],

                    [250, 50],

                    [100, 33],

                    [330, 95],

                    [410, 12],

                    [475, 44],

                    [25, 67],

                    [85, 21],

                    [220, 88],

                            ];

    var xScale = d3.scaleLinear() //create a linear scale for the x-axis

                .domain([d3.min(dataset, function(d){

                    return d[0];

                }),

                d3.max(dataset, function(d){

                    return d[0];

                })])

                .range([padding, w - padding]);

    var yScale = d3.scaleLinear() //create a linear scale for the y-axis

                .domain([d3.min(dataset, function(d){

                    return d[1];

                }),

                d3.max(dataset, function(d){

                    return d[1];

                })])

                .range([h - padding, padding]);

    var svg = d3.select("#scatter-pot")

                .append("svg")

                .attr("width", w)

                .attr("height", h);

    svg.selectAll("circle")

        .data(dataset)

        .enter()

        .append("circle")

        .attr("cx", function(d, i){

            return xScale(d[0]); //assign the cx to the second value of the 2d array

        })

        .attr("cy", function(d){

            return yScale(d[1]); //assign cy to the second value of the 2d array

        })

        .attr("r", 5) //the radius

        .attr("fill", function(d){

            if (d[0] === 500 && d[1] === 90) {

                return "red";

            } else {

                return "grey";

            }

        });

    svg.selectAll("text")

        .data(dataset)

        .enter()

        .append("text")

        .text(function(d){

            return d[0] + "," + d[1]; //display the x and y values

        })

        .attr("x", function(d){

            return xScale(d[0]); // set the x position to the x value from data

        })

        .attr("y", function(d){

            return yScale(d[1]) - 11; // adjust the y position to center the text vertically

        })

        .attr("text-anchor", "middle") // center the text horizontally

        .attr("fill", "green");

}

window.onload = init;

3.2.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8"/>

    <meta name="description" content="Data Visualisation"/>

    <meta name="keyword" content="HTML, CSS"/>

    <meta name="author" content="Minh Nguyen"/>

    <title>Task 3.2 Axis Chart</title>

    <script src="https://d3js.org/d3.v5.min.js"></script>

    <script src="3.2.js"></script>

    <link rel="stylesheet" href="3.2.css">

</head>

<body>

    <h1>Drawing with Data - Scatter Plot</h1>

    <div id="scatter-pot"></div>

    <br>

    <bf></bf>

    <footer style="color:grey">COS30045 Data Visualisation<br>Minh Nguyen</footer>

</body>

</html>

3.2.css

.axis path,

.axis line {

    stroke: teal;

    shape-rendering: crispEdges;

}

.axis text {

    font-family: sans-serif;

    font-weight: bold;

    font-size: 14px;

    fill: teal;

}

p {

    color: olive

}

text{

    fill: olive

}

3.2.js

function init()

{

    var w = 600; // subtracting 40px for padding

    var h = 500; // subtracting 20px for padding

    var padding = 50;

    var dataset = [

                    [5, 20],

                    [500, 90],

                    [250, 50],

                    [100, 33],

                    [330, 95],

                    [410, 12],

                    [475, 44],

                    [25, 67],

                    [85, 21],

                    [220, 88],

                            ];

    var xScale = d3.scaleLinear() //create a linear scale for the x-axis

                .domain([d3.min(dataset, function(d){

                    return d[0];

                }),

                d3.max(dataset, function(d){

                    return d[0];

                })])

                .range([padding, w - padding]);

    var yScale = d3.scaleLinear() //create a linear scale for the y-axis

                .domain([d3.min(dataset, function(d){

                    return d[1];

                }),

                d3.max(dataset, function(d){

                    return d[1];

                })])

                .range([h - padding, padding]);

    var xAxis = d3.axisBottom().ticks(10).scale(xScale); //create the x-axis

    var yAxis = d3.axisLeft().ticks(10).scale(yScale); //create the y-axis

    //create the SVG element

    var svg = d3.select("#scatter-pot")

                .append("svg")

                .attr("width", w)

                .attr("height", h);

    svg.selectAll("circle")

        .data(dataset)

        .enter()

        .append("circle")

        .attr("cx", function(d, i){

            return xScale(d[0]); //assign the cx to the second value of the 2d array

        })

        .attr("cy", function(d){

            return yScale(d[1]); //assign cy to the second value of the 2d array

        })

        .attr("r", 5) //the radius

        .attr("fill", function(d){

            if (d[0] === 500 && d[1] === 90) {

                return "red"; //for the highest value

            } else {

                return "grey";

            }

        });

    svg.selectAll("text")

        .data(dataset)

        .enter()

        .append("text")

        .text(function(d){

            return d[0] + "," + d[1]; //display the x and y values

        })

        .attr("x", function(d){

            return xScale(d[0]); // assign the x position to the x value in data

        })

        .attr("y", function(d){

            return yScale(d[1]) - 10; // adjust the y position to center the text vertically

        })

        .attr("text-anchor", "middle") // center the text horizontally

        .attr("fill", "green");

        //Append the x and y axis to the SVG element as groups

    svg.append("g")

        .attr("transform", "translate(0, "+ (h - padding + 10) +")") //translating them to the appropriate positions

        .call(xAxis);

    svg.append("g")

        .attr("transform", "translate(" + (padding) + ", 10)")

        .call(yAxis);

}

window.onload = init;

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động