<!-- <div id="my\_dataviz"> </div> -->

<div id="my\_data"> </div>

<!-- Create a div where the graph will take place -->

<!-- <div id="dataviz"></div>

< div id="my\_dataviz"> </div>

<script>

// set the dimensions and margins of the graph

// You can change these values these are just sample values given

var margin = {top: 20, right: 50, bottom: 50, left: 60},

width = 460 - margin.left - margin.right,

height = 400 - margin.top - margin.bottom;

// append the svg object to the body of the page

var svg = d3.select("#dataviz")

.append("svg")

.attr("width", width + margin.left + margin.right)

.attr("height", height + margin.top + margin.bottom)

.append("g")

.attr("transform",

"translate(" + margin.left + "," + margin.top + ")");

// uncomment the function and complete this function to plot required graphs

// d3.csv("https://github.com/vega/vega/blob/main/docs/data/seattle-weather.csv", function(data) {

d3.csv("seattle-weather.csv", function(data) {

console.log(data[0])

// X axis: scale and draw:

var x = d3.scaleLinear()

// d3.max(data, function(d) { return + d.wind }

// .domain([0, 1000]) // can use this instead of 1000 to have the max of data: d3.max(data, function(d) { return +d.price })

// var xaxis= d3.max(data, function(d) { return + d.wind })

.domain([0, 10])

.range([0, width]);

svg.append("g")

.attr("transform", "translate(0," + height + ")")

.call(d3.axisBottom(x));

// set the parameters for the histogram

var histogram = d3.histogram()

.value(function(d) { return d.wind; }) // I need to give the vector of value

.domain(x.domain()) // then the domain of the graphic

.thresholds(x.ticks(25)); // then the numbers of bins

// And apply this function to data to get the bins

var bins = histogram(data);

// Y axis: scale and draw:

var y = d3.scaleLinear()

.range([height, 0]);

y.domain([0, d3.max(bins, function(d) { return d.length; })]); // d3.hist has to be called before the Y axis obviously

svg.append("g")

.call(d3.axisLeft(y));

// append the bar rectangles to the svg element

svg.append("text")

.attr("class", "x label")

.attr("text-anchor", "end")

.attr("x", width)

.attr("y", height - 6)

.text("Group");

svg.append("text")

.attr("class", "y label")

.attr("text-anchor", "end")

.attr("x", 35)

.attr("y", 6)

.attr("dy", ".75em")

// .attr("transform", "rotate(-90)")

.text("wind");

svg.selectAll("rect")

.data(bins)

.enter()

.append("rect")

.attr("x", 1)

.attr("transform", function(d) { return "translate(" + x(d.x0) + "," + y(d.length) + ")"; })

.attr("width", function(d) { return x(d.x1) - x(d.x0) -1 ; })

.attr("height", function(d) { return height - y(d.length); })

.style("fill", "#69b3a2")

});

</script> -->

<!--

<script>

// set the dimensions and margins of the graph

var width = 450

height = 450

margin = 40

// The radius of the pieplot is half the width or half the height (smallest one). I subtract a bit of margin.

var radius = Math.min(width, height) / 2 - margin

var arcGenerator = d3.arc()

.innerRadius(0)

.outerRadius(radius+200)

// append the svg object to the div called 'my\_dataviz'

var svg1 = d3.select("#my\_dataviz")

.append("svg")

.attr("width", width)

.attr("height", height)

.append("g")

.attr("transform", "translate(" + width / 2 + "," + height / 2 + ")");

var data = { sun: 640, snow : 26, rain :641, fog:101, drizzle:53}

// Compute the position of each group on the pie:

// set the color scale

var color = d3.scaleOrdinal()

.domain(data)

.range(["#98abc5", "#8a89a6", "#7b6888", "#6b486b", "#a05d56"])

var pie = d3.pie()

.value(function(d) {return d.value; })

var data\_ready = pie(d3.entries(data))

// Build the pie chart: Basically, each part of the pie is a path that we build using the arc function.

svg1

.selectAll('whatever')

.data(data\_ready)

.enter()

.append('path')

.attr('d', d3.arc()

.innerRadius(0)

.outerRadius(radius)

)

.attr('fill', function(d){ return(color(d.data.key)) })

.attr("stroke", "black")

.style("stroke-width", "2px")

.style("opacity", 0.7)

svg1

.selectAll('mySlices')

.data(data\_ready)

.enter()

.append('text')

.text(function(d){ return d.data.key ; })

.attr("transform", function(d) { return "translate(" + arcGenerator.centroid(d) + ")"; })

.style("text-anchor", "middle")

.style("font-size", 10)

var arc = d3.arc()

.innerRadius(0)

.outerRadius(radius-5)

svg1

.selectAll('mySlices')

.data(data\_ready)

.enter()

.append('text')

.text(function(d){ return (Math.round(d.data.value/1461 \* 100)).toFixed(0) + '%' ; })

.attr("transform", function(d) { return "translate(" + arc.centroid(d) + ")"; })

.style("text-anchor", "middle")

.style("font-size", 10)

</script> -->