Line Chart

HTML

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Our First Line Chart</title>

</head>

<body>

<div id="chart-container"></div>

<!-- Include the D3.js library -->

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

<!-- Include your custom script file -->

<script src="scriptstart.js"></script>

</body>

</html>

Java Script

// Set dimensions and margins for the chart

const margin = { top: 70, right: 30, bottom: 40, left: 80 };

const width = 1200 - margin.left - margin.right;

const height = 500 - margin.top - margin.bottom;

// Set up the x and y scales

const x = d3.scaleTime()

.range([0, width]);

const y = d3.scaleLinear()

.range([height, 0]);

// Create the SVG element and append it to the chart container

const svg = d3.select("#chart-container")

.append("svg")

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

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

.append("g")

.attr("transform", `translate(${margin.left},${margin.top})`);

// Create a fake dataset

const dataset = [

{ date: new Date("2022-01-01"), value: 200 },

{ date: new Date("2022-02-01"), value: 250 },

{ date: new Date("2022-03-01"), value: 180 },

{ date: new Date("2022-04-01"), value: 300 },

{ date: new Date("2022-05-01"), value: 280 },

{ date: new Date("2022-06-01"), value: 220 },

{ date: new Date("2022-07-01"), value: 300 },

{ date: new Date("2022-08-01"), value: 450 },

{ date: new Date("2022-09-01"), value: 280 },

{ date: new Date("2022-10-01"), value: 600 },

{ date: new Date("2022-11-01"), value: 780 },

{ date: new Date("2022-12-01"), value: 320 }

];

// Define the x and y domains

x.domain(d3.extent(dataset, d => d.date));

y.domain([0, d3.max(dataset, d => d.value)]);

// Add the x-axis

svg.append("g")

.attr("transform", `translate(0,${height})`)

.call(d3.axisBottom(x)

.ticks(d3.timeMonth.every(1))

.tickFormat(d3.timeFormat("%b %Y")));

// Add the y-axis

svg.append("g")

.call(d3.axisLeft(y))

// Create the line generator

const line = d3.line()

.x(d => x(d.date))

.y(d => y(d.value));

// Add the line path to the SVG element

svg.append("path")

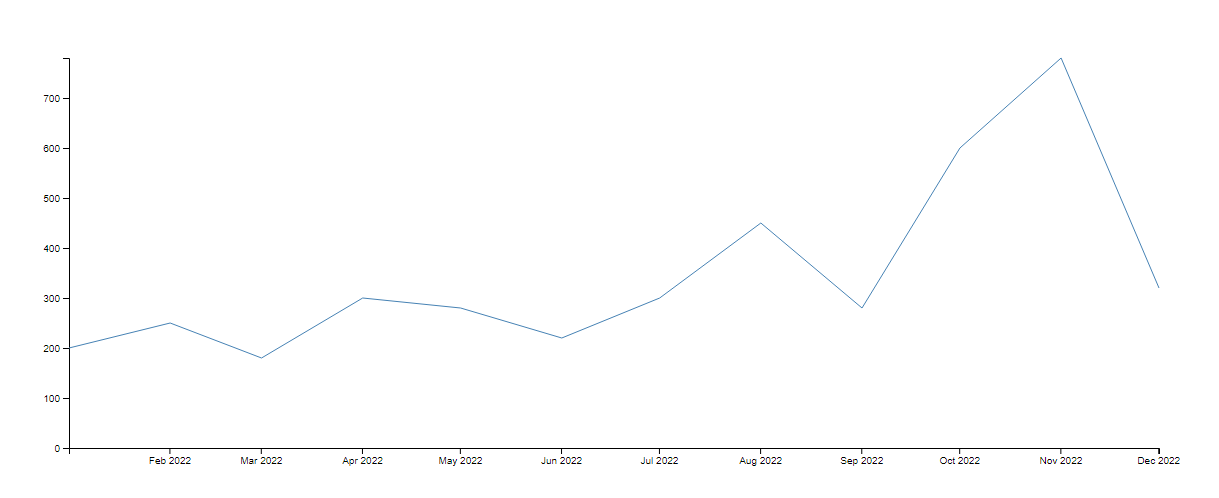
.datum(dataset)

.attr("fill", "none")

.attr("stroke", "steelblue")

.attr("stroke-width", 1)

.attr("d", line);



2) Scatter plot

<!DOCTYPE html>

<meta charset="utf-8">

<!-- Load d3.js -->

<script src="https://d3js.org/d3.v4.js"></script>

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

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

<script>

// set the dimensions and margins of the graph

var margin = {top: 10, right: 30, bottom: 30, 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("#my\_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 + ")");

//Read the data

d3.csv("https://raw.githubusercontent.com/holtzy/data\_to\_viz/master/Example\_dataset/2\_TwoNum.csv", function(data) {

// Add X axis

var x = d3.scaleLinear()

.domain([0, 4000])

.range([ 0, width ]);

svg.append("g")

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

.call(d3.axisBottom(x));

// Add Y axis

var y = d3.scaleLinear()

.domain([0, 500000])

.range([ height, 0]);

svg.append("g")

.call(d3.axisLeft(y));

// Add dots

svg.append('g')

.selectAll("dot")

.data(data)

.enter()

.append("circle")

.attr("cx", function (d) { return x(d.GrLivArea); } )

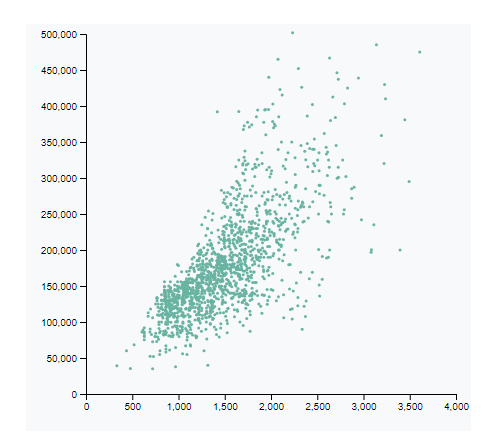
.attr("cy", function (d) { return y(d.SalePrice); } )

.attr("r", 1.5)

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

})

</script>



3) Bar chart

<svg id="chart"></svg>

var data = [4, 5, 10, 16, 23, 35];

var width = 500;

var barHeight = 20;

var x = d3.scale.linear()

.domain([0, d3.max(data)])

.range([0, width]);

var chart = d3.select('#chart')

.attr('width', width)

.attr('height', barHeight \* data.length);

var bar = chart.selectAll('g')

.data(data)

.enter()

.append('g')

.attr('transform', function(d, i){

return 'translate(0,' + barHeight \* i + ')';

});

bar.append('rect')

.attr('width', x)

.attr('height', barHeight - 1);

bar.append('text')

.attr('x', function(d){

return x(d) - 3;

})

.attr('y', barHeight / 2)

.attr('dy', '.35em')

.text(function(d){

return d;

});

