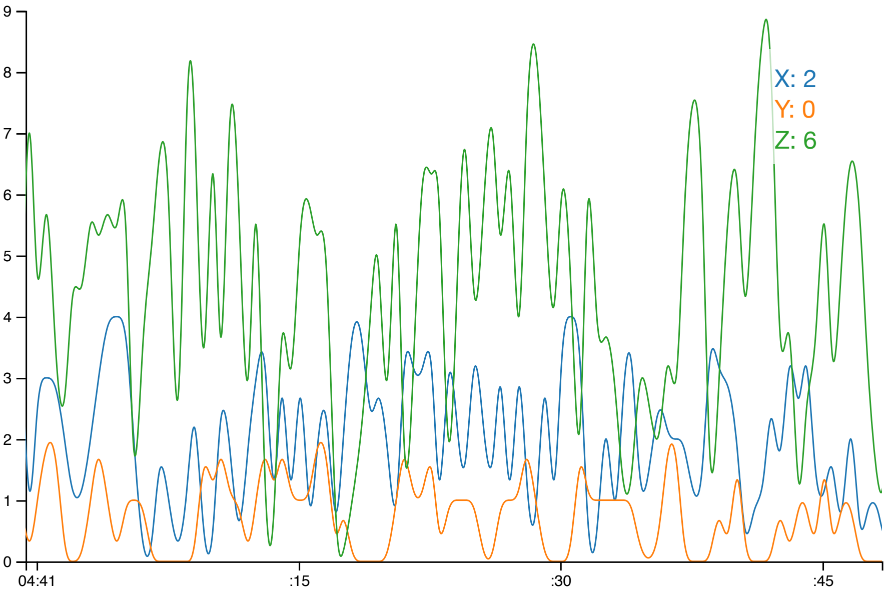
Patrick Sier's Block fbf9317b31f070fd540c5523fef167ac
Updated July 28, 2017

Popular / About

Real-Time Line Chart



Open

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Realtime Data Line Graph</title>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <meta charset='utf-8' />
    <script src="https://d3js.org/d3.v4.min.js"></script>
    <style>
    body {
        font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
    }
    #chart {
      max-width: 600px;
      max-height: 400px;
    }
    </style>
  </head>
  <body>
    <div id="chart"></div>
    <script src="script.js"></script>
    <script>
    var lineArr = [];
    var MAX_LENGTH = 100;
    var duration = 500;

    var chart = realTimeLineChart();

    function randomNumberBounds(min, max) {
      return Math.floor(Math.random() * max) + min;
    }

    function seedData() {
      var now = new Date();
      for (var i = 0; i < MAX_LENGTH; ++i) {
        lineArr.push({
          time: new Date(now.getTime() - ((MAX_LENGTH - i) * duration)),
          x: randomNumberBounds(0, 5),
          y: randomNumberBounds(0, 2.5),
          z: randomNumberBounds(0, 10)
        });
      }
    }

    function updateData() {
      var now = new Date();

      var lineData = {
        time: now,
        x: randomNumberBounds(0, 5),
        y: randomNumberBounds(0, 2.5),
        z: randomNumberBounds(0, 10)
      };
      lineArr.push(lineData);

      if (lineArr.length > 30) {
        lineArr.shift();
      }
      d3.select("#chart").datum(lineArr).call(chart);
    }

    function resize() {
      if (d3.select("#chart svg").empty()) {
        return;
      }
      chart.width(+d3.select("#chart").style("width").replace(/(px)/g, ""));
      d3.select("#chart").call(chart);
    }

    document.addEventListener("DOMContentLoaded", function() {
      seedData();
      window.setInterval(updateData, 500);
      d3.select("#chart").datum(lineArr).call(chart);
      d3.select(window).on('resize', resize);
    });
    </script>
</body>
</html>
```

script.js

```
function realTimeLineChart() {
  var margin = {top: 20, right: 20, bottom: 20, left: 20},
      width = 600,
      height = 400,
      duration = 500,
      color = d3.schemeCategory10;

  function chart(selection) {
    // Based on https://bl.ocks.org/mbostock/3884955
    selection.each(function(data) {
      data = ["x", "y", "z"].map(function(c) {
        return {
          label: c,
          values: data.map(function(d) {
            return {time: +d.time, value: d[c]};
          })
        };
      });

      var t = d3.transition().duration(duration).ease(d3.easeLinear),
          x = d3.scaleTime().rangeRound([0, width-margin.left-margin.right]),
          y = d3.scaleLinear().rangeRound([height-margin.top-margin.bottom, 0]),
          z = d3.scaleOrdinal(color);

      var xMin = d3.min(data, function(c) { return d3.min(c.values, function(d) { return d.time; })
      var xMax = new Date(new Date(d3.max(data, function(c) {
        return d3.max(c.values, function(d) { return d.time; })
      })).getTime() - (duration*2));

      x.domain([xMin, xMax]);
      y.domain([
        d3.min(data, function(c) { return d3.min(c.values, function(d) { return d.value; })}),
        d3.max(data, function(c) { return d3.max(c.values, function(d) { return d.value; })})
      ]);
      z.domain(data.map(function(c) { return c.label; }));

      var line = d3.line()
        .curve(d3.curveBasis)
        .x(function(d) { return x(d.time); })
        .y(function(d) { return y(d.value); });

      var svg = d3.select(this).selectAll("svg").data([data]);
      var gEnter = svg.enter().append("svg").append("g");
      gEnter.append("g").attr("class", "axis x");
      gEnter.append("g").attr("class", "axis y");
      gEnter.append("defs").append("clipPath")
          .attr("id", "clip")
        .append("rect")
          .attr("width", width-margin.left-margin.right)
          .attr("height", height-margin.top-margin.bottom);
      gEnter.append("g")
          .attr("class", "lines")
          .attr("clip-path", "url(#clip)")
        .selectAll(".data").data(data).enter()
          .append("path")
            .attr("class", "data");

      var legendEnter = gEnter.append("g")
        .attr("class", "legend")
        .attr("transform", "translate(" + (width-margin.right-margin.left-75) + ",25)");
      legendEnter.append("rect")
        .attr("width", 50)
        .attr("height", 75)
        .attr("fill", "#ffffff")
        .attr("fill-opacity", 0.7);
      legendEnter.selectAll("text")
        .data(data).enter()
        .append("text")
          .attr("y", function(d, i) { return (i*20) + 25; })
          .attr("x", 5)
          .attr("fill", function(d) { return z(d.label); });

      var svg = selection.select("svg");
      svg.attr('width', width).attr('height', height);
      var g = svg.select("g")
        .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

      g.select("g.axis.x")
        .attr("transform", "translate(0," + (height-margin.bottom-margin.top) + ")")
        .transition(t)
```

```
        .call(d3.axisBottom(x).ticks(5));
      g.select("g.axis.y")
        .transition(t)
        .attr("class", "axis y")
        .call(d3.axisLeft(y));

      g.select("defs clipPath rect")
        .transition(t)
        .attr("width", width-margin.left-margin.right)
        .attr("height", height-margin.top-margin.right);

      g.selectAll("g path.data")
        .data(data)
        .style("stroke", function(d) { return z(d.label); })
        .style("stroke-width", 1)
        .style("fill", "none")
        .transition()
        .duration(duration)
        .ease(d3.easeLinear)
        .on("start", tick);

      g.selectAll("g .legend text")
        .data(data)
        .text(function(d) {
          return d.label.toUpperCase() + ": " + d.values[d.values.length-1].value;
        });

      // For transitions https://bl.ocks.org/mbostock/1642874
      function tick() {
        d3.select(this)
          .attr("d", function(d) { return line(d.values); })
          .attr("transform", null);

        var xMinLess = new Date(new Date(xMin).getTime() - duration);
        d3.active(this)
            .attr("transform", "translate(" + x(xMinLess) + ",0)")
          .transition()
            .on("start", tick);
      }
    });
  }

  chart.margin = function(_) {
    if (!arguments.length) return margin;
    margin = _;
    return chart;
  };

  chart.width = function(_) {
    if (!arguments.length) return width;
    width = _;
    return chart;
  };

  chart.height = function(_) {
    if (!arguments.length) return height;
    height = _;
    return chart;
  };

  chart.color = function(_) {
    if (!arguments.length) return color;
    color = _;
    return chart;
  };

  chart.duration = function(_) {
    if (!arguments.length) return duration;
    duration = _;
    return chart;
  };

  return chart;
}
```

LICENSE