

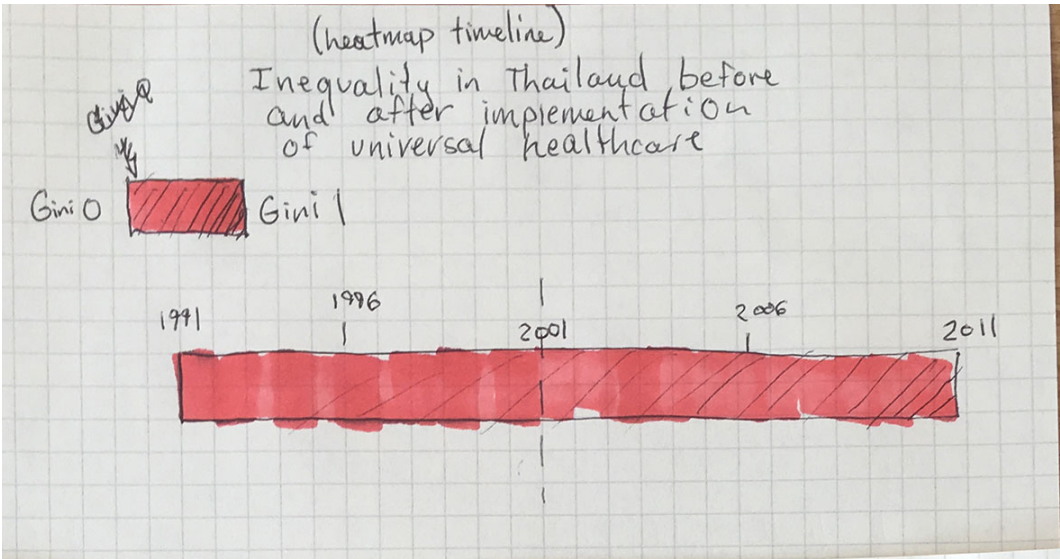
First sketches

Research question:

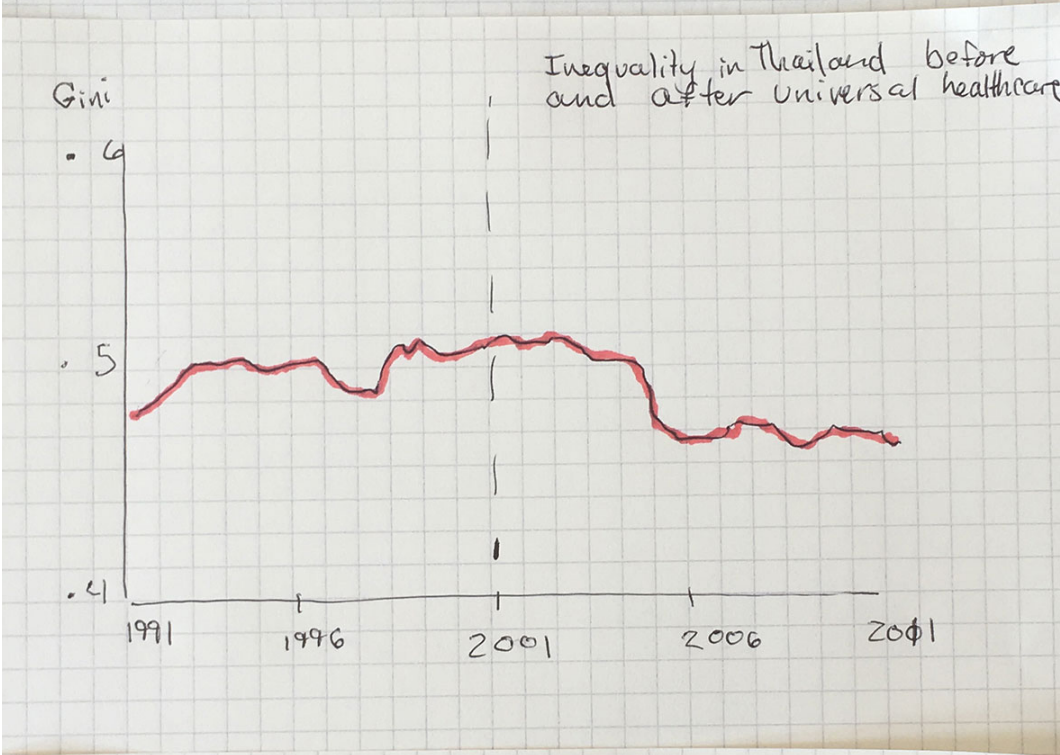
Is there a visible relationship between the implementation of universal healthcare and inequality in Thailand?

Goal:

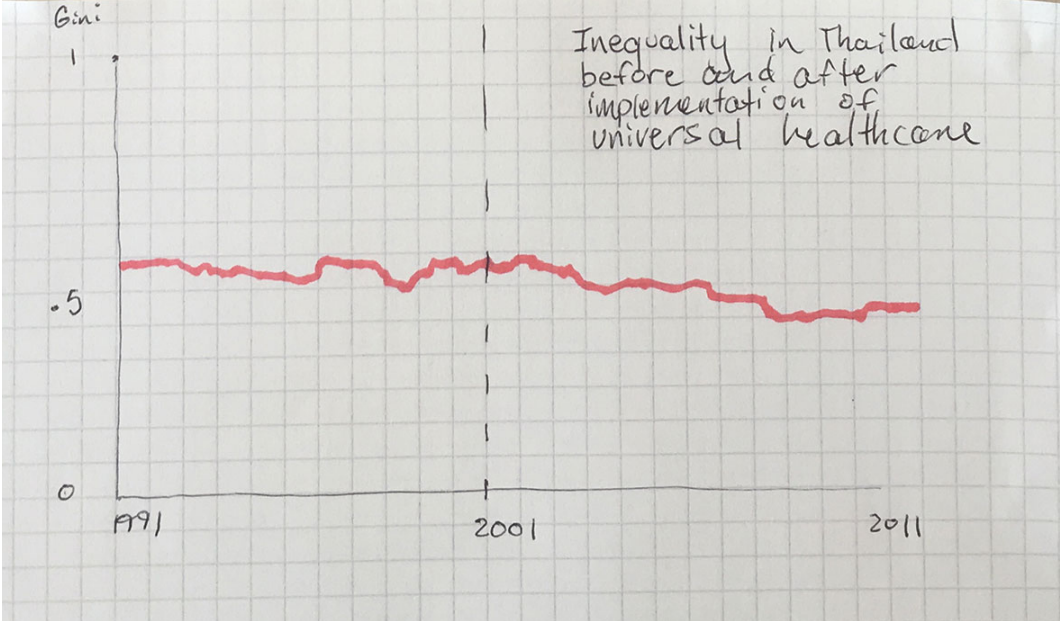
Show the change in Gini ten years before and ten years after the 2001 implementation of universal healthcare in Thailand



Idea 1:
Heatmap Timeline



Idea 2:
Line chart with zoomed Gini



Idea 3:
Line chart with normal Gini

Attempt 1

I wanted to start by calling and logging the data, then build from there.

```
var countryOne = [];
var countryTwo = [];

// load data and parse it down to Thailand rows

d3.csv('/project-1/data/wiid.csv', function(d) {
  return {
    // make sure numbers are read as numbers and not as strings
    Country : d.Country,
    Year : +d.Year,
    Gini : +d.Gini
  };
  // provide accessor function to d3.js to return individual objects in data array
}, function (data) {
  for (var i = 0; i<data.length; i++){
    var allData = data[i];
    if (allData.Country == "Thailand" && allData.Year >= 1991){
      countryOne.push(allData);
    }
  }
});

console.log (countryOne);
```

```
▶0: {Country: "Thailand", Year: 1992, Gini: 42.7}
▶1: {Country: "Thailand", Year: 1992, Gini: 57}
▶2: {Country: "Thailand", Year: 1992, Gini: 51.5}
▶3: {Country: "Thailand", Year: 1992, Gini: 44.5}
▶4: {Country: "Thailand", Year: 1994, Gini: 43.2}
▶5: {Country: "Thailand", Year: 1994, Gini: 57.1}
▶6: {Country: "Thailand", Year: 1994, Gini: 43.1}
▶7: {Country: "Thailand", Year: 1996, Gini: 43.3}
▶8: {Country: "Thailand", Year: 1996, Gini: 55.5}
▶9: {Country: "Thailand", Year: 1996, Gini: 42.9}
▶10: {Country: "Thailand", Year: 1998, Gini: 43.2}
▶11: {Country: "Thailand", Year: 1998, Gini: 55.6}
▶12: {Country: "Thailand", Year: 1998, Gini: 42.1}
▶13: {Country: "Thailand", Year: 1999, Gini: 42.3}
▶14: {Country: "Thailand", Year: 1999, Gini: 56}
▶15: {Country: "Thailand", Year: 1999, Gini: 44.4}
▶16: {Country: "Thailand", Year: 2000, Gini: 43.9}
▶17: {Country: "Thailand", Year: 2000, Gini: 42.83}
▶18: {Country: "Thailand", Year: 2001, Gini: 41.9}
```

← Product

```
d3.csv('/project-1/data/wiid.csv', function(error, data) {
  if (error) throw error;
  // svg(data);
  data.forEach(function(d, i) {
    if (d.Country == "Thailand" && d.Year>= 1991){
      i=i-7825;
      console.log (i);
      svg.append('text')
        .text (d.Year)
        .attr('x', i * 40)
        .attr('y', innerHeight/1.1)
        .style('text-anchor', 'float: left')
        .style('fill', 'black')
        .style("font", "9px times");
    }
  });
});
```

Input

I was able to parse and log the data so I pushed it to an array hoping to manipulate it. I wasn't able to access it from the array other than to log it.

Input

I was able to access individual rows of data from within my forEach conditional. From there I appended the year text to my svg. The iterator was huge, though, so I manipulated i before using it to space out my dates.

Unfortunately I wasn't able to append any other elements to data using this method

← Product

Attempt 2

```
var margin = {top: 20, right: 20, bottom: 30, left: 40},
    width = 960 - margin.left - margin.right,
    height = 500 - margin.top - margin.bottom;

var xValue = function(d) {
  if (d.Country == "Thailand" && d.Year>= 1993){
    return d.Year
  }
}; // data -> value
xScale = d3.scaleLinear().range([0, width]), // value -> display
xMap = function(d) {
  if (d.Country == "Thailand" && d.Year>= 1993){
    return xScale(xValue(d));
  }
}; // data -> display
xAxis = d3.axisBottom().scale(xScale);

var yValue = function(d) {
  if (d.Country == "Thailand" && d.Year>= 1993){
    return d.Gini;
  }
}; // data -> value
yScale = d3.scaleLinear().range([height, 0]), // value -> display
yMap = function(d) { return yScale(yValue(d));}, // data -> display
yAxis = d3.axisLeft().scale(yScale);

// setup fill color
var cValue = 'red';

// add the graph canvas to the body of the webpage
var svg = d3.select("#lineChart").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 + ")");

// add the tooltip area to the webpage
var tooltip = d3.select("body").append("div")
  .attr("class", "tooltip")
  .style("opacity", 0);

d3.csv('/project-1/data/wiid.csv', function(error, data) {
  if (error) throw error;
  // svg(data);
  data.forEach(function(d, i) {

    if (d.Country == "Thailand" && d.Year>= 1993){
      // x-axis
      svg.append("g")
        .attr("class", "x axis")
        .attr("transform", "translate(0," + height + ")")
        .call(xAxis)
        .append("text")
        .attr("class", "label")
        .attr("x", width)
        .attr("y", -6)
        .style("text-anchor", "end")
        .text("hello");

      // y-axis
      svg.append("g")
        .attr("class", "y axis")
        .call(yAxis)
        .append("text")
        .attr("class", "label")
        .attr("transform", "rotate(-90)")
        .attr("y", 6)
        .attr("dy", ".71em")
        .style("text-anchor", "end")
        .text(d.Gini);

    }

  })
});
```

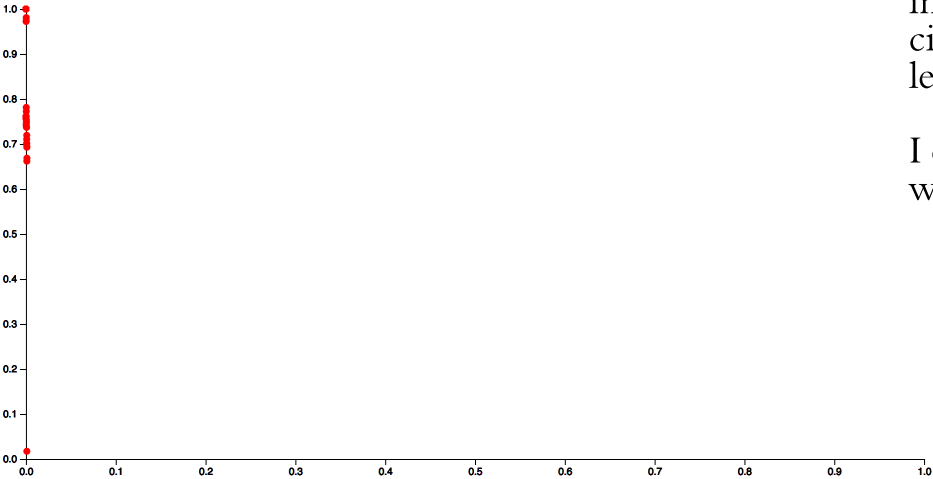
Input

I began with a Mike Bostock scatterplot template and added in conditionals to parse the data.

Product

I was able to connect the circles visually on my chart to Gini index, but thousands of empty circles were being sent to my svg leading to a really slow load time.

I experimented for a while but wasn't able to make this work.



Attempt 3

```
var svgContainer = d3.select("body")
    .append("svg")
    .attr("width", window.innerWidth)
    .attr("height", window.innerHeight);

var backGround = svgContainer.append("rect")
    .attr("width", window.innerWidth)
    .attr("height", window.innerHeight);

var circle = svgContainer.append("circle")
    .attr("cx", 30)
    .attr("cy", 30)
    .attr("r", 2);
circle.style("fill", "red");

var leftBound = svgContainer.append("line")
    .attr("x1", window.innerWidth/8)
    .attr("y1", window.innerHeight - 100)
    .attr("x2", window.innerWidth/8)
    .attr("y2", window.innerHeight/14)
    .attr("stroke", "white")
    .attr("stroke-width", .5);

var bottomBound = svgContainer.append("line")
    .attr("x1", window.innerWidth-50)
    .attr("y1", window.innerHeight -100)
    .attr("x2", window.innerWidth/8)
    .attr("y2", window.innerHeight-100)
    .attr("stroke", "white")
    .attr("stroke-width", .5);

var countryOne = □;
var countryOneYear = □;
var countryOneGini = □;

d3.request("/project-1/data/wiid.csv")
    .mimeType("text/csv")
    .response(function (xhr) { return d3.csvParse(xhr.responseText); })
    .get(function(data) {
        for (var i = 0; i<data.length; i++){
            var allData = data[i];
            if (allData.Country == "Thailand" && allData.Year >= 1993){
                var text = svgContainer.append('text')
                    .text (allData.Year)
                    .attr('x', window.innerWidth/8+i*.0013)
                    .attr('y', window.innerHeight -90*i*.00013)
                    .style('text-anchor', 'float: left')
                    .style('fill', 'white')
                    .style("font", "9px times");
            }
        }
    });
```

Input

I manually scripted all visual elements first and then used my first, must successful code in attempt to connect the data

Product

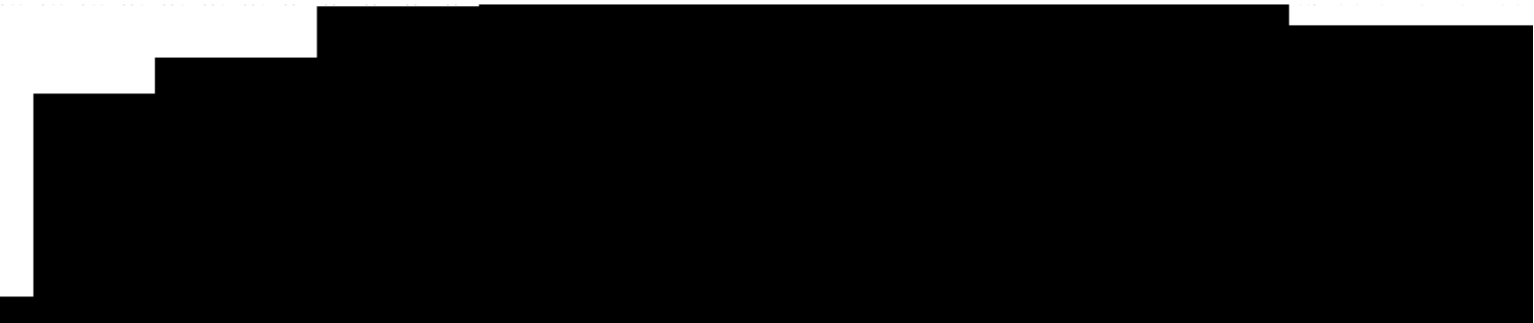
As in my first attempt, I was able to connect the text to the data but not anything else.

Attempt 4

I manually converted my data to a .JSON file and worked directly off of Daniel's template

Product

Everything connected properly, but I had run out of time to finish this



Conclusions

Working within templates can be more difficult than hand coding. Looking back my most successful attempts were hand coded

I'm not comfortable using callback functions and that really limits my understanding and use of Javascript