```
<!DOCTYPE html>
<head>
   <meta charset="utf-8">
     <script type="text/javascript" src="d3.js" rel="javascript"></script>
     <link rel="stylesheet" href="gini.css" type="text/css" media="screen" />
     <link rel="stylesheet" href="wave_colours.css" type="text/css" media="screen" />
<!--     <link rel="stylesheet" href="axis.css" type="text/css" media="screen" /> -->
<script type="text/javascript">
var mi_scale;
var dhi_scale;
var path;
var adjustMargins=0;
var countryDisplayed =[];
var neutral_line_displayed = -1;
var gini_mi_extent = [];
var gini_dhi_extent = [];
var xLimit=[0,1];
var yLimit=[0,1];
var mi_axis = d3.svg.axis();
var dhi_axis = d3.svg.axis();

var container_dimensions = {width: 680, height: 600},
       margins = {top: 10, right: 20, bottom: 30, left: 60},
       chart_dimensions = {
          width: container_dimensions.width - margins.left - margins.right,
          height: container_dimensions.height - margins.top - margins.bottom
       };



function drawNeutralLine(){

      if ( neutral_line_displayed == 1 ) {
            path = d3.select('g#neutral_line.Line_np');
            path.remove();
            neutral_line_displayed = -1;
         // Remove dots
         //path = null;
      } else {

            var neutral_gini_mi = [];
         var neutral_gini_dhi = [];


            var lbNeutral = d3.max([d3.min(yLimit),d3.min(xLimit)]);
            var ubNeutral = d3.min([d3.max(xLimit),d3.max(yLimit)]);

            for (i=0;i<101;i++) {
                  neutral_gini_mi.push(lbNeutral+(.01*i)*(ubNeutral-lbNeutral));
```

```
                neutral_gini_dhi.push(lbNeutral+(.01*i)*(ubNeutral-lbNeutral));
            }

            var line = d3.svg.line()
                .x(function(neutral_gini_mi){return mi_scale(neutral_gini_mi);})
                .y(function(neutral_gini_dhi){return dhi_scale(neutral_gini_dhi);})
                .interpolate("linear");

            var g = d3.select('#chart')
                .append('g')
                .attr('id','neutral_line')
                .attr('class','Line_np');

              g.append('path')
                .attr('d', line(neutral_gini_dhi));

            neutral_line_displayed=1;
          // Add dots etc.
       }
}

function computeNewAxisLimits(d,i) {

  if (countryDisplayed.length == 0) {
     alert("You must add at least one \n series to the plot before you zoom");
     }
   else {
          var gini_mi_extent_temp=[];
          var gini_dhi_extent_temp=[];


//            var id = countryDisplayed[i];

//          d3.json('/data/lis-gini.json',function(data){
//            filtered_data = data.filter(function(d){return
//          for (i=0;i<countryDisplayed.length;i++)
//             {
//                (d.line_id === id) ||
//             }
//          });
//            gini_mi_extent_temp = d3.extent(filtered_data,function(d) {return d.gini_mi});
//            if (gini_mi_extent.length == 0) {
//                gini_mi_extent = gini_mi_extent_temp;
//                }
//            else {
//                if (gini_mi_extent_temp[0] < gini_mi_extent[0] ) {
//                   gini_mi_extent[0] = gini_mi_extent_temp[0];
//                   }
//                if (gini_mi_extent_temp[1] > gini_mi_extent[1] ) {
```

```
//                      gini_mi_extent[1] = gini_mi_extent_temp[1];
//                      }
//          }


//          gini_dhi_extent_temp = d3.extent(filtered_data,function(d) {return d.gini_dhi});
//          if (gini_dhi_extent.length == 0) {
//                gini_dhi_extent = gini_dhi_extent_temp;
//                }
//          else {
//                if (gini_dhi_extent_temp[0] < gini_dhi_extent[0] ) {
//                   gini_dhi_extent[0] = gini_dhi_extent_temp[0];
//                   }
//                if (gini_dhi_extent_temp[1] > gini_dhi_extent[1] ) {
//                   gini_dhi_extent[1] = gini_dhi_extent_temp[1];
//                   }
//          }



//        mi_scale = d3.scale.linear()
//                .range([0, chart_dimensions.width])
//                .domain([gini_mi_extent[0],gini_mi_extent[1]]);


//        dhi_scale = d3.scale.linear()
//                .range([0, chart_dimensions.height])
//                .domain([gini_dhi_extent[1], gini_dhi_extent[0]]);

}
}


function adjustMarginsF(){

   adjustMargins=1;
   d3.json('/data/lis_gini_recent.json', function(d){


        xLimit=[.27,.58];
        yLimit=[.15,.38];

   mi_scale = d3.scale.linear()
   .range([0, chart_dimensions.width])
   .domain([xLimit[0],xLimit[1]] );

   dhi_scale = d3.scale.linear()
   .range([0, chart_dimensions.height])
   .domain([yLimit[1],yLimit[0]]);
```

```
        mi_axis.scale(mi_scale).orient("bottom").ticks(10);
        dhi_axis.scale(dhi_scale).orient("left").ticks(10);

        d3.select(".x.axis")
           .transition()
           .duration(1000)
           .call(mi_axis);

        d3.select(".y.axis")
           .transition()
           .duration(1000)
           .call(dhi_axis);

        for (i=0; i < countryDisplayed.length;i++) {
           get_wave_data(d,countryDisplayed[i]);
           }

});
}

 function get_wave_data(d,i){
    var id = d.line_id;
    var wv = d3.select('#'+id);
   if (adjustMargins == 0) {
        if (wv.empty()){
           d3.json('/data/lis-gini.json',function(data){
                filtered_data = data.filter(function(d){return d.line_id === id;});
                draw_wave(filtered_data, id);
                countryDisplayed.push(id)
           });
        } else {
           wv.remove();
           var index = countryDisplayed.indexOf(id);
           countryDisplayed.splice(index,1); }
         } else {
        // draw new line
        d3.json('/data/lis-gini.json',function(data){
           filtered_data = data.filter(function(d){return d.line_id === id;});
           draw_wave_transition(filtered_data,id);
        // do a transition


   });

        }
   }

function add_label(circle, d, i) {
```

```
    d3.select(circle)
      .transition()
      .attr('r',11);

    d3.select('#' + d.line_id).append('text')
    .text(d.line_id.split('_')[1])
    .attr('text-anchor','middle')
    .style("dominant-baseline","central")
    .attr('x', mi_scale(d.gini_mi))
    .attr('y', dhi_scale(d.gini_dhi))
    .attr('class','linelabel')
    .style('opacity',0)
    .style('fill','white')
    .transition()
    .style('opacity',1);
    }

function draw_wave(data, id){

var line = d3.svg.line()
    .x(function(d){return mi_scale(d.gini_mi)})
    .y(function(d){return dhi_scale(d.gini_dhi)})
    .interpolate("linear")

var g = d3.select('#chart')
    .append('g')
    .attr('id',id)
    .attr('class','wave ' + id);

g.append('path')
    .attr('d', line(data));

g.selectAll('circle')
    .data(data)
    .enter()
    .append('circle')
    .attr('cx', function(d) { return mi_scale(d.gini_mi);})
    .attr('cy', function(d) { return dhi_scale(d.gini_dhi);})
    .attr('r',0);

var enter_duration = 1000;

g.selectAll('circle')
    .transition()
    .delay(function(d,i) { return i / data.length * enter_duration; })
    .attr('r', 5)
    .each('end', function(d,i) {
          if (i === data.length-1) {
```

```
            add_label(this,d)
        }
    });

g.selectAll('circle')
    .on('mouseover', function(d) {
        d3.select(this)
            .transition().attr('r',9)
    })
    .on('mouseout', function(d,i){
        if (i !== data.length-1) {
            d3.select(this).transition().attr('r',5)
        }
    });

 g.selectAll('circle')
 .on('mouseover.tooltip', function(d){
     d3.select("text." + d.line_id).remove()
     d3.select('#chart')
        .append('text')
        .text("gini mi=" + d.gini_mi + ", " + "gini dhi=" + d.gini_dhi)
        .attr('x', mi_scale(d.gini_mi) + 10)
        .attr('y', dhi_scale(d.gini_dhi) - 10)
        .attr('class', d.line_id)
 })
 .on('mouseout.tooltip', function(d){
     d3.select("text." + d.line_id)
        .transition()
        .duration(500)
        .style('opacity',0)
        .attr('transform','translate(10, -10)')
        .remove()
 });

}

function draw_wave_transition(data, id){

var line = d3.svg.line()
    .x(function(d){return mi_scale(d.gini_mi)})
    .y(function(d){return dhi_scale(d.gini_dhi)})
    .interpolate("linear");

    d3.select('wave ' + id)
    .transition()
    .duration(1000)
    .attr('d',line(data));
```

```
g.selectAll('circle')
    .data(data)
    .enter()
    .append('circle')
    .attr('cx', function(d) { return mi_scale(d.gini_mi)})
    .attr('cy', function(d) { return dhi_scale(d.gini_dhi)})
    .attr('r',0);



g.selectAll('circle')
    .transition()
    .delay(function(d,i) { return i / data.length * enter_duration; })
    .attr('r', 5)
    .each('end', function(d,i) {
          if (i === data.length-1) {
             add_label(this,d)
          }
    });

g.selectAll('circle')
    .on('mouseover', function(d) {
          d3.select(this)
              .transition().attr('r',9)
    })
    .on('mouseout', function(d,i){
          if (i !== data.length-1) {
             d3.select(this).transition().attr('r',5)
          }
    });

 g.selectAll('circle')
 .on('mouseover.tooltip', function(d){
     d3.select("text." + d.line_id).remove()
     d3.select('#chart')
        .append('text')
        .text("gini mi=" + d.gini_mi + ", " + "gini dhi=" + d.gini_dhi)
        .attr('x', mi_scale(d.gini_mi) + 10)
        .attr('y', dhi_scale(d.gini_dhi) - 10)
        .attr('class', d.line_id)
 })
 .on('mouseout.tooltip', function(d){
     d3.select("text." + d.line_id)
        .transition()
        .duration(500)
        .style('opacity',0)
        .attr('transform','translate(10, -10)')
        .remove()
 });
```

```
}

function draw(data) {
    "use strict";

// Draw the

    xLimit=[0,1];
    yLimit=[0,1];
    mi_scale = d3.scale.linear()
    .range([0, chart_dimensions.width])
          .domain([xLimit[0],xLimit[1]] );


    dhi_scale = d3.scale.linear()
    .range([0, chart_dimensions.height])
    .domain([yLimit[1],yLimit[0]]);


    mi_axis.scale(mi_scale).orient("bottom").ticks(5);
    dhi_axis.scale(dhi_scale).orient("left").ticks(5);

    var g = d3.select('#wave')
    .append('svg')
    .attr('width', container_dimensions.width)
    .attr('height', container_dimensions.height)
    .append("g")
    .attr("transform", "translate(" + margins.left + "," + margins.top + ")")
    .attr("id","chart");

    g.append("g")
    .attr("class","x axis")
    .attr("transform", "translate( 0 ," + chart_dimensions.height + ")")
    .call(mi_axis);

    d3.select('.x.axis')
    .append('text')
    .text('market gini index')
    .attr('x', 250)
    .attr('y',25);

    // svg.append("text")
    //      .attr("class","x label")
    //      .attr("text-anchor", "end")
    //      .attr("x", width)
    //      .attr("y", height)
```

```
    g.append("g")
    .attr("class", "y axis")
    .call(dhi_axis);

    d3.select('.y.axis')
    .append('text')
    .text('post tax and transfer gini index')
    .attr('transform', "rotate (-270, 0, 0)")
          .attr('x', 100)
    .attr('y',50);

    var key_items = d3.select('#key')
      .selectAll('div')
      .data(data)
      .enter()
      .append('div')
        .attr('class','key_line')
        .attr('id',function(d){return d.line_id+"_key"});

    key_items.append('div')
        .attr('id', function(d){return 'key_square_' + d.line_id})
        .attr('class', function(d){return 'key_square ' + d.line_id});

    key_items.append('div')
        .attr('class','key_label')
        .text(function(d){return d.line_name});

    d3.selectAll('.key_line')
        .on('click', get_wave_data);


        }

</script>
</head>
<body>
  <div id="wave"></div>
    <div id="key"></div>
<div id="policy_neutral">
 <input name="policyButton"
              type="button"
             value="Policy Neutral Line"
             onclick="drawNeutralLine()" />
</div>
<div id="margins">
    <input name="marginButton"
              type="button"
                    value="Zoom in/out"
             onclick="adjustMarginsF()" />
```

```
</div>
<div id="poverty">
   <input name="povertyButton"
              type="button"
             value="Add poverty"
             onclick="addPoverty()" />
</div>



<script>
     d3.json('/data/lis_gini_recent.json', draw);
</script>
</body>
</html>
```