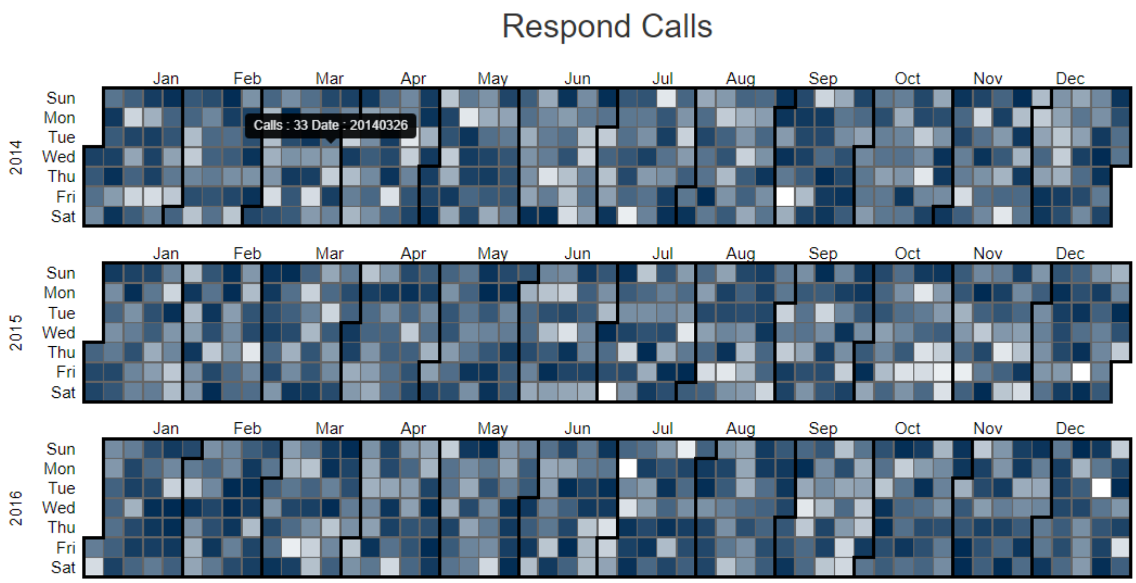

HTML:

```
<!DOCTYPE html>
<head>
    <title>Respond Calls</title>

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

    <link href="./bootstrap.min.css" rel="stylesheet" />
    <script type="text/javascript" src="./bootstrap.min.js"></script>
    <script type="text/javascript" src="./d3.v3.min.js"></script>
    <script src="https://d3js.org/d3-queue.v3.min.js"></script>

    <style>
    .calender-map {
      font: 10px sans-serif;
      shape-rendering: crispEdges;
    }
    .day {
      stroke: #666;
    }
    .month {
      fill: none;
      stroke: #000;
      stroke-width: 2px;
    }
    .RdYlGn .q0-11{fill:rgb(165,0,38)}
    .RdYlGn .q1-{fill:rgb(215,48,39)}
    .RdYlGn .q2-11{fill:rgb(244,109,67)}
    .RdYlGn .q3-11{fill:rgb(253,174,97)}
    .RdYlGn .q4-11{fill:rgb(254,224,139)}
    .RdYlGn .q5-11{fill:rgb(255,255,191)}
    .RdYlGn .q6-11{fill:rgb(217,239,139)}
    .RdYlGn .q7-11{fill:rgb(166,217,106)}
    .RdYlGn .q8-11{fill:rgb(102,189,99)}
    .RdYlGn .q9-11{fill:rgb(26,152,80)}
    .RdYlGn .q10-11{fill:rgb(0,104,55)}
    </style>
</head>
<body>
<h2 style="text-align: center;"> Respond Calls</h2>
<div class="calender-map"></div>

<!--<script type="text/javascript"
src="../../js/test_files/calendermap.js"></script>-->
<script type="text/javascript" src="./calendermap.js"></script>
</body></html>
```

JavaScript:

```
var q = d3.queue()
      .defer(d3.csv,"./calls.csv")
     // .defer(d3.json,"./calls.csv")
      .await(CreateCalender);

var width = 900,
    height = 105,
    cellSize = 12; // cell size
    week_days = ['Sun','Mon','Tue','Wed','Thu','Fri','Sat']
    month =
['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']
    //csv
    //respondSQL;

var day = d3.time.format("%w"),
    week = d3.time.format("%U"),
    percent = d3.format(".1%"),
    format = d3.time.format("%Y%m%d");
    parseDate = d3.time.format("%Y%m%d").parse;

var color = d3.scale.linear().range(["white", '#002b53'])
    .domain([0, 1])


var svg = d3.select(".calender-map").selectAll("svg")
    .data(d3.range(2014, 2017))
//    .data(d3.range(2014,2018))
  .enter().append("svg")
    .attr("width", '100%')
    .attr("data-height", '0.5678')
    .attr("viewBox",'0 0 900 105')
    .attr("class", "RdYlGn")
  .append("g")
    .attr("transform", "translate(" + ((width - cellSize * 53) / 2) + "," +
(height - cellSize * 7 - 1) + ")");

svg.append("text")
    .attr("transform", "translate(-38," + cellSize * 3.5 + ")rotate(-90)")
    .style("text-anchor", "middle")
    .text(function(d) { return d; });

for (var i=0; i<7; i++)
{
svg.append("text")
    .attr("transform", "translate(-5," + cellSize*(i+1) + ")")
    .style("text-anchor", "end")
    .attr("dy", "-.25em")
    .text(function(d) { return week_days[i]; });
 }

var rect = svg.selectAll(".day")
    .data(function(d) { return d3.time.days(new Date(d, 0, 1), new Date(d +
1, 0, 1)); })
  .enter()
    .append("rect")
    .attr("class", "day")
    .attr("width", cellSize)
    .attr("height", cellSize)
    .attr("x", function(d) { return week(d) * cellSize; })
    .attr("y", function(d) { return day(d) * cellSize; })
    .attr("fill",'#fff')
    .datum(format);

var legend = svg.selectAll(".legend")
      .data(month)
    .enter().append("g")
      .attr("class", "legend")
      .attr("transform", function(d, i) { return "translate(" + (((i+1) *
50)+8) + ",0)"; });

legend.append("text")
   .attr("class", function(d,i){ return month[i] })
   .style("text-anchor", "end")
   .attr("dy", "-.25em")
   .text(function(d,i){ return month[i] });

svg.selectAll(".month")
    .data(function(d) { return d3.time.months(new Date(d, 0, 1), new Date(d +
1, 0, 1)); })
  .enter().append("path")
    .attr("class", "month")
    .attr("id", function(d,i){ return month[i] })
    .attr("d", monthPath);




function CreateCalender(error, respond1) {//, respondSQL1) {
    if (error) {
        console.log(error);
    }
    //respondSQL = respondSQL1;
    respond = respond1;

    csv = respond1; //d3.merge([respond, respondSQL]);
  csv.forEach(function(d) {
    d.Comparison_Type = parseInt(d.EVENTS);
     //tformat = d3.time.format("%m/%d/%Y");
     tformat = d3.time.format("%Y%m%d");
    var mysql = d3.time.format("%Y-%m-%d");
    var mapformat = d3.time.format("%Y%m%d");
    //console.log(d);
    if (d.DOS.indexOf("-") < 0) {
        d.tmp = tformat.parse(d.DOS);
    } else {
        d.tmp = mysql.parse(d.DOS);
    }

    d.Date = format(d.tmp);
  });

  Comparison_Type_Max = d3.max(csv, function(d) { return
d.Comparison_Type; });

   data = d3.nest()
    .key(function(d) { return d.Date; })
  .rollup(function(d) { return {"Comparison_Type":
Math.sqrt(d[0].Comparison_Type / Comparison_Type_Max), "total_calls":
d[0].Comparison_Type, "DOS": d[0].tmp} })
    .map(csv);

  rect.filter(function(d) { return d in data; })
      .attr("fill", function(d) { return color(data[d].Comparison_Type); })
    //  .attr("data-title", function(d) { return "Calls :
"+Math.round(data[d]*100)});
    .attr("data-title", function(d) { return "Calls : " + data[d].total_calls
+ " Date : " + tformat(data[d].DOS) });
    $("rect").tooltip({container: 'body', html: true, placement:'top'});
}

function numberWithCommas(x) {
    x = x.toString();
    var pattern = /(-?\d+)(\d{3})/;
    while (pattern.test(x))
        x = x.replace(pattern, "$1,$2");
    return x;
}

function monthPath(t0) {
  var t1 = new Date(t0.getFullYear(), t0.getMonth() + 1, 0),
      d0 = +day(t0), w0 = +week(t0),
      d1 = +day(t1), w1 = +week(t1);
  return "M" + (w0 + 1) * cellSize + "," + d0 * cellSize
      + "H" + w0 * cellSize + "V" + 7 * cellSize
      + "H" + w1 * cellSize + "V" + (d1 + 1) * cellSize
      + "H" + (w1 + 1) * cellSize + "V" + 0
      + "H" + (w0 + 1) * cellSize + "Z";
}
```