data-vis\helper-functions.js

```
// ---------------------------------------------------------------------
// Data processing helper functions.
// ---------------------------------------------------------------------
function sum(data) {
  var total = 0;

  // Ensure that data contains numbers and not strings.
  data = stringsToNumbers(data);

  for (let i = 0; i < data.length; i++) {
    total = total + data[i];
  }

  return total;
}

function mean(data) {
  var total = sum(data);

  return total / data.length;
}

function sliceRowNumbers (row, start=0, end) {
  var rowData = [];

  if (!end) {
    // Parse all values until the end of the row.
    end = row.arr.length;
  }

  for (i = start; i < end; i++) {
    rowData.push(row.getNum(i));
  }

  return rowData;
}

function stringsToNumbers (array) {
  return array.map(Number);
}

// ---------------------------------------------------------------------
// Plotting helper functions
// ---------------------------------------------------------------------

//------------------------------ START NEW CODE ---------------------------------------
-//
function drawAxis(layout, colour=0, threeAxis=false) {
  stroke(color(colour));
  strokeWeight(1);

  if (threeAxis) {
```

```
  // x-axis
  line(layout.leftMargin,
       layout.bottomMargin,
       layout.rightMargin,
       layout.bottomMargin);

  // y-axis
  line(layout.leftMargin,
       layout.topMargin,
       layout.leftMargin,
       layout.bottomMargin);

  // y2-axis
  line(layout.rightMargin,
       layout.topMargin,
       layout.rightMargin,
       layout.bottomMargin);

//------------------------------ END NEW CODE ----------------------------------------//

  }
  else {
   // x-axis
  line(layout.leftMargin,
       layout.bottomMargin,
       layout.rightMargin,
       layout.bottomMargin);

  // y-axis
  line(layout.leftMargin,
       layout.topMargin,
       layout.leftMargin,
       layout.bottomMargin);
  }
}

function drawAxisLabels(xLabel, y1Label, y2Label="", layout) {
  fill(0);
  noStroke();
  textAlign('center', 'center');
  textFont(robotoFont);
  textSize(16);

  // Draw x-axis label.
  text(xLabel,
       (layout.plotWidth() / 2) + layout.leftMargin,
       layout.bottomMargin + (layout.marginSize * 0.25));

  // Draw y-axis label.
  push();
  translate(layout.leftMargin - (layout.marginSize * 0.3),
            layout.bottomMargin / 1.45);
  rotate(- PI / 2);
  text(y1Label, 0, 0);
```

```
106   pop();
107
108   if (y2Label != "") {
109     // Draw y2-axis label.
110     push();
111     translate(layout.rightMargin - (layout.marginSize * 0.3),
112               layout.bottomMargin / 1.45);
113     rotate(- PI / 2);
114     text(y2Label, 0, 80);
115     pop();
116   }
117 }
118
119 function drawYAxisTickLabels(min, max, layout, mapFunction,
120                              decimalPlaces) {
121   // Map function must be passed with .bind(this).
122   var range = max - min;
123   var yTickStep = range / layout.numYTickLabels;
124
125   fill(0);
126   noStroke();
127   textAlign('right', 'center');
128   textFont(robotoFont);
129   textSize(16);
130
131   // Draw all axis tick labels and grid lines.
132   for (i = 0; i <= layout.numYTickLabels; i++) {
133     var value = min + (i * yTickStep);
134     var y = mapFunction(value);
135
136     // Add tick label.
137     text(value.toFixed(decimalPlaces),
138          layout.leftMargin - layout.pad,
139          y);
140
141     if (layout.grid) {
142       // Add grid line.
143       stroke(200);
144       strokeWeight(1);
145       line(layout.leftMargin, y, layout.rightMargin, y);
146     }
147   }
148 }
149
150 //------------------------------ START NEW CODE ---------------------------------------
    -//
151 function drawY2AxisTickLabels(min, max, layout, mapFunction,
152                              decimalPlaces) {
153   // Map function must be passed with .bind(this).
154   var range = max - min;
155   var yTickStep = range / layout.numYTickLabels;
156
157   fill(0);
158   noStroke();
```

```
  textAlign('right', 'center');
  textFont(robotoFont);
  textSize(16);

  // Draw all axis tick labels
  for (i = 0; i <= layout.numYTickLabels; i++) {
    var value = min + (i * yTickStep);
    var y = mapFunction(value);

    // Add tick label.
    text(value.toFixed(decimalPlaces),
         layout.rightMargin + 4 * layout.pad,
         y);
  }
}
//------------------------------ END NEW CODE ----------------------------------------//

function drawXAxisTickLabel(value, layout, mapFunction) {
  // Map function must be passed with .bind(this).
  var x = mapFunction(value);

  fill(0);
  noStroke();
  textAlign('center', 'center');
  textFont(robotoFont);

  // Add tick label.
  text(value,
       x,
       layout.bottomMargin + layout.marginSize / 8);

  if (layout.grid) {
    // Add grid line.
    stroke(220);
    strokeWeight(1);
    line(x,
         layout.topMargin,
         x,
         layout.bottomMargin);
  }
}

//------------------------------ START NEW CODE ---------------------------------------
-//
function draw2Legend(xPos, yPos, labelText1, labelText2) {
  textFont(robotoFont);
  textAlign(LEFT, CENTER);
  textSize(16);
  noFill();
  noStroke();

  fill('#C8102E');
  rect(xPos, yPos, 15, 15, 3);
  fill(0);
```

```
  text(labelText1, xPos + 25, yPos + 6);

  fill('#002147');
  rect(xPos + 120, yPos, 15, 15, 3);
  fill(0);
  text(labelText2, xPos + 145, yPos + 6);
}

function drawLegend(xPos, yPos, labels) {
  // Space per block
  let spacing = 145;
  let boxSize = 15;

  // Font
  textFont(robotoFont);
  textAlign(LEFT, CENTER);
  textSize(16);
  noStroke();

  for (let i = 0; i < labels.length; i++) {
    //
    let currentX = xPos + i * spacing;

    //
    fill(labels[i].color);
    rect(currentX, yPos, boxSize, boxSize, 3);

    //
    fill(0);
    text(labels[i].text, currentX + boxSize + 10, yPos + boxSize / 2);
  }
}
//------------------------------ END NEW CODE ----------------------------------------//

```