data-vis\pie-chart.js

```
function PieChart(x, y, diameter) {

  this.x = x;
  this.y = y;
  this.diameter = diameter;
  this.labelSpace = 30;

  this.get_radians = function(data) {
    var total = sum(data);
    var radians = [];

    for (let i = 0; i < data.length; i++) {
      radians.push((data[i] / total) * TWO_PI);
    }

    return radians;
  };

  this.draw = function(data, labels, colours, title) {

    // Test that data is not empty and that each input array is the
    // same length.
    if (data.length == 0) {
      alert('Data has length zero!');
    } else if (![labels, colours].every((array) => {
      return array.length == data.length;
    })) {
      alert(`Data (length: ${data.length})
Labels (length: ${labels.length})
Colours (length: ${colours.length})
Arrays must be the same length!`);
    }

    // https://p5js.org/examples/form-pie-chart.html

    var angles = this.get_radians(data);
    var lastAngle = 0;
    var colour;

    for (var i = 0; i < data.length; i++) {
      if (colours) {
        colour = colours[i];
      } else {
        colour = map(i, 0, data.length, 0, 255);
      }

      fill(colour);
      stroke(0);
      strokeWeight(1);

      arc(this.x, this.y,
```

```
52          this.diameter, this.diameter,
53          lastAngle, lastAngle + angles[i] + 0.001); // Hack for 0!
54
55      //------------------------------------------------------------------------ START NEW CODE ------------------------------------------------------------------------//
56      // Creating a variable "d" to calculate the distance between two points. Here, it's the distance between the mouse (point 1) and the center of the pie chart (point 2).
57      // Structure: dist(x1, y1, x2, y2), where x1/y1 are coordenates of the first point and x2/y2 coordenats of the second point.
58      let d = dist(mouseX, mouseY, this.x, this.y);
59      let mouse_pie_angle = null;  // ✅ DECLARADA FORA PARA SER USADA DEPOIS
60
61      //
62      if (d < this.diameter / 2) { // Divided by two since radius = diameter/2.
63        // atan2(y, x) is from arctangente, it calculates the angle of a vector between 2 points, where x is the distance in x and y the distance in y. IT'S THE SIZE OF A VECTOR.
64        // It returns an angle in radius from -PI to PI (-180º to +180º)
65        mouse_pie_angle = atan2(mouseY - this.y, mouseX - this.x);
66
67        if (mouse_pie_angle < 0) {
68          mouse_pie_angle += TWO_PI;
69        }
70      }
71
72      if (mouse_pie_angle !== null && mouse_pie_angle >= lastAngle && mouse_pie_angle < lastAngle + angles[i]) {
73        let [r, g, b] = colours[i].levels;
74        let valueText = data[i].toFixed(2) + "%";
75
76        textSize(14);
77        textAlign('left', 'top');
78        textLeading(18);
79
80        let textW = textWidth(valueText) + 10;
81        let textH = 25;
82
83        // Horizontal position
84        let tooltipX = (mouseX + 10 + textW > width)
85                     ? mouseX - textW - 10
86                     : mouseX + 10;
87
88        // Vertical position
89        let tooltipY = mouseY - textH - 10;
90        if (tooltipY < 0) {
91          tooltipY = mouseY + 10;
92        }
93
94        // Background
95        fill(255, 255, 255, 230);
96        stroke(0);
97        strokeWeight(1);
98        rect(tooltipX, tooltipY, textW, textH);
99
100       // Text
```

```
      fill(0);
      noStroke();
      text(valueText, tooltipX + 5, tooltipY + 5);
    }

      if (labels) {
        this.makeLegendItem(labels[i], i, colour);
      }

      lastAngle += angles[i];
    }

    if (title) {
      noStroke();
      textAlign('center', 'center');
      textSize(24);
      text(title, this.x, this.y - this.diameter * 0.6);
    }
  };

  this.makeLegendItem = function(label, i, colour) {
    var x = this.x + 50 + this.diameter / 2;
    var y = this.y + (this.labelSpace * i) - this.diameter / 3;
    var boxWidth = this.labelSpace / 2;
    var boxHeight = this.labelSpace / 2;

    fill(colour);
    rect(x, y, boxWidth, boxHeight);

    fill('black');
    noStroke();
    textAlign('left', 'center');
    textSize(12);
    text(label, x + boxWidth + 10, y + boxWidth / 2);
  };
}
```