data-vis\pay-gap-by-job-2017.js

```
function PayGapByJob2017() {

  // Offset for the x/y-axis.
  this.offsetX = 550;
  this.offsetY = 300;
  this.scale =  0.7;

  // Name for the visualisation to appear in the menu bar.
  this.name = 'Pay GAP per Job';

  // Each visualisation must have a unique ID with no special
  // characters.
  this.id = 'pay-gap-by-job-2017';

  // Property to represent whether data has been loaded.
  this.loaded = false;

  // Graph properties.
  this.pad = 20;
  this.padX = this.pad + this.offsetX;

  this.dotSizeMin = 15;
  this.dotSizeMax = 40;

  // Preload the data. This function is called automatically by the
  // gallery when a visualisation is added.
  this.preload = function() {
    var self = this;
    this.data = loadTable(
      './data/pay-gap/occupation-hourly-pay-by-gender-2017.csv', 'csv', 'header',
      // Callback function to set the value
      // this.loaded to true.
      function(table) {
        self.loaded = true;
      });

  };

  this.setup = function() {
  };

  this.destroy = function() {
  };

  this.draw = function() {
    if (!this.loaded) {
      console.log('Data not yet loaded');
      return;
    }

    // Draw the axes.
```

```
52      this.addAxes();
53
54      // Get data from the table object.
55      var jobs = this.data.getColumn('job_subtype');
56      var propFemale = this.data.getColumn('proportion_female');
57      var payGap = this.data.getColumn('pay_gap');
58      var numJobs = this.data.getColumn('num_jobs');
59
60      // Convert numerical data from strings to numbers.
61      propFemale = stringsToNumbers(propFemale);
62      payGap = stringsToNumbers(payGap);
63      numJobs = stringsToNumbers(numJobs);
64
65      // Set ranges for axes.
66      //
67      // Use full 100% for x-axis (proportion of women in roles).
68      var propFemaleMin = 0;
69      var propFemaleMax = 100;
70
71      // For y-axis (pay gap) use a symmetrical axis equal to the
72      // largest gap direction so that equal pay (0% pay gap) is in the
73      // centre of the canvas. Above the line means men are paid
74      // more. Below the line means women are paid more.
75      var payGapMin = -20;
76      var payGapMax = 20;
77
78      // Find smallest and largest numbers of people across all
79      // categories to scale the size of the dots.
80      var numJobsMin = min(numJobs);
81      var numJobsMax = max(numJobs);
82
83      fill(255);
84      stroke(0);
85      strokeWeight(1);
86
87      for (i = 0; i < this.data.getRowCount(); i++) {
88        // Draw an ellipse for each point.
89        // x = propFemale
90        // y = payGap
91        // size = numJobs
92      ellipse(
93        ( map(propFemale[i], propFemaleMin, propFemaleMax,
94              this.pad, width - this.pad)
95          + this.offsetX
96        ) * this.scale,
97
98        ( map(payGap[i], payGapMin, payGapMax,
99              height - this.pad, this.pad)
100         + this.offsetY
101       ) * this.scale,
102
103       map(numJobs[i], numJobsMin, numJobsMax,
104           this.dotSizeMin, this.dotSizeMax)
105       * this.scale
```

```
    );
    }
  };

  this.addAxes = function () {
    stroke(200);

    // Add vertical line.
    line(
      (width/2 + this.offsetX) * this.scale,
      (this.pad           + this.offsetY) * this.scale,
      (width/2 + this.offsetX) * this.scale,
      (height - this.pad  + this.offsetY) * this.scale
    );

    // Add horizontal line.
    line(
      (this.pad + this.offsetX)          * this.scale,
      (height/2 + this.offsetY)          * this.scale,
      (width   - this.pad + this.offsetX) * this.scale,
      (height/2 + this.offsetY)          * this.scale
    );
  };
}

```