Parallel Coordinates and Scatter Plots

Exercise 1

- Parallel coordinates
  - For multi-dimensional data
  - Very useful, but a little bit complicated
    - Each dimension is visualized as a vertical axis.
    - Values in each dimension are be appropriately mapped.
    - Each data point is a polyline to connect its values on individual dimension axis.
- Scatter plots
  - Only for two dimensions.

Parallel Coordinates

- Key steps in the drawpc() function
  - Prepare for individual dimensions
    - Their horizontal locations
    - Their domains (min and max values)
  - Draw individual polylines
    - Go through each record
      - Go through each dimension
        - Find its value
        - Get x (based on dimension) and y (based on value) coordinates
      - Draw an SVG line (a path shape)
  - Draw dimension axes and labels
- You can use the codes for future parallel coordinates visualization, with some minor modifications based on the data you have (e.g., data attribute names).

Scatter Plot with Filled Circles

- Circle is a basic shape in D3.
- Required attributes
  - cx: the x coordinate of the center
  - cy: the y coordinate of the center
  - r: radius

```
.attr("cx", x)
.attr("cy", y)
.attr("r", r)
```

- Styles:
  - stroke, stroke-width, fill

Rect vs. Circle in D3

```
//building bars
chart.selectAll(".bar")
    .data(bardata)
    .enter().append("rect")
    .attr("class", "bar")
    .attr("x", function(d) { return x(d.name); })
    .attr("y", function(d) { return y(d.value); })
    .attr("height", function(d) { return height - y(d.value); })
    .attr("width", x.rangeBand())


  //draw a dot
  var dot = svg.append("g")
        .append("circle")
        .attr("class", "dot")
        .attr("cx", function(d) { return x(cars[i].year); })
        .attr("cy", function(d) { return y(cars[i].power); })
        .attr("idx", i)
        .attr("r", 3)
        .style("fill", "black")
```

Exercise 2: Combining Two Graphs

- Shrink the graphs first.
  - Hint: change the range of the x domain.
- Move the scatter plot to the right.

- Combine two JS documents
  - Function and other statement related to the definition of required elements, such as axis domain variables.
    - Be careful of duplicated variable names.

Exercise 3: Making Two Graphs Coordinated

- Highlighting behaviors are local in each function.
- New approach:
  - Mouse events to identify which data element is concerned (e.g., the index of the data element in the array).
    - Define an attribute for each graph element (polyline or dot) to record its index. .attr("index", i)
  - Mouse events to call functions to process polylines and dots with the identified index.
    - Define arrays for polyline and dot objects

    dots[i].style(…)

    polyLines[i].style(…)