# D3: Brushing and Linking

Oct. 23, 2019

Brushing and Linking as user interaction method

- Highlighting associations/connections between related items
  - Different attributes/variables of the same data case
  - The same data case but with multiple views
- Selecting or highlighting a data case in one view selects/highlights the same data case in the other views
  - "Brushing": selecting some subset of the data
 - "Linking": highlighting the same subset in multiple views

Example

- Vis showing many attributes from a dataset of cars, with each pair of numeric attributes plotted in their own scatterplot (a scatterplot matrix, or "splom")
- Each data case also has a color corresponding to the number of cylinders in that car's engine
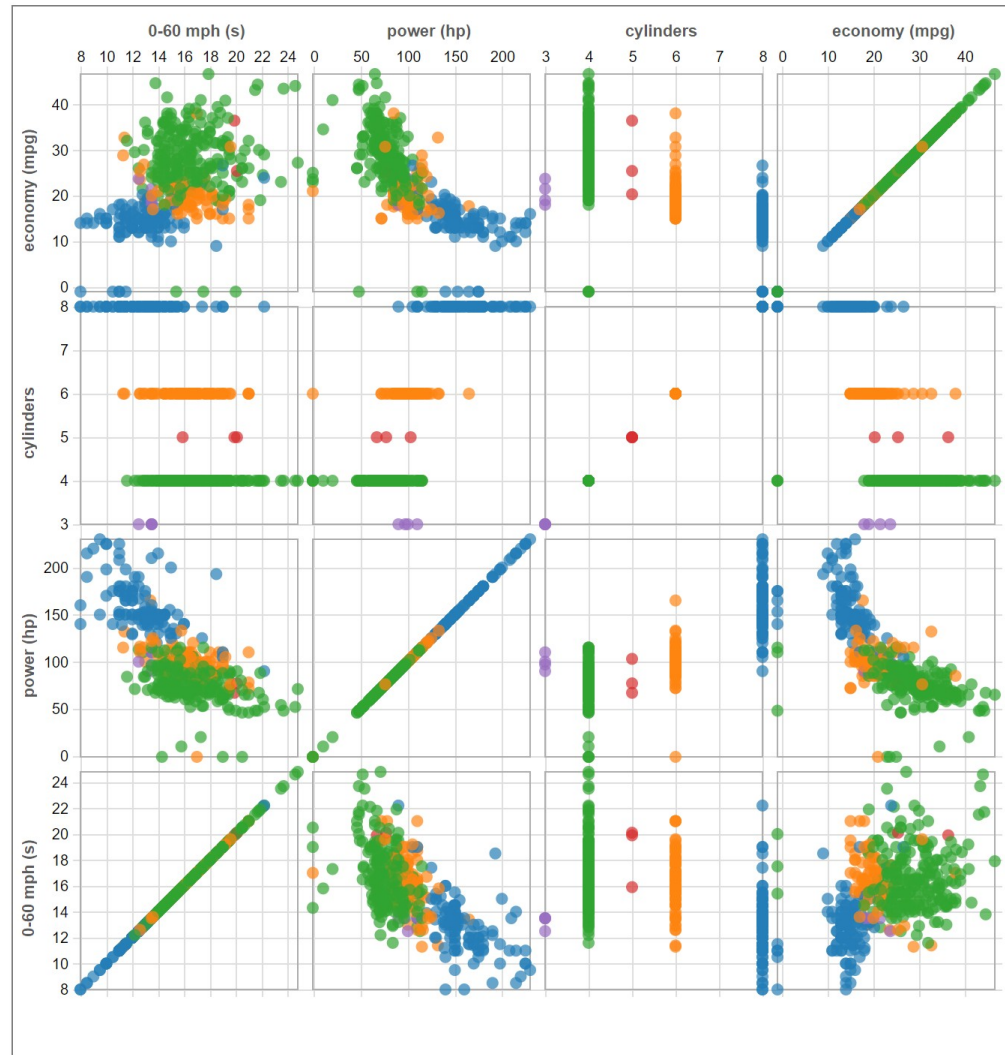
Example

- In the "# cylinders vs. economy" chart, we select all of the data cases where cylinders = 8, and then the same data cases are highlighted in all 16 charts, while the rest are greyed out
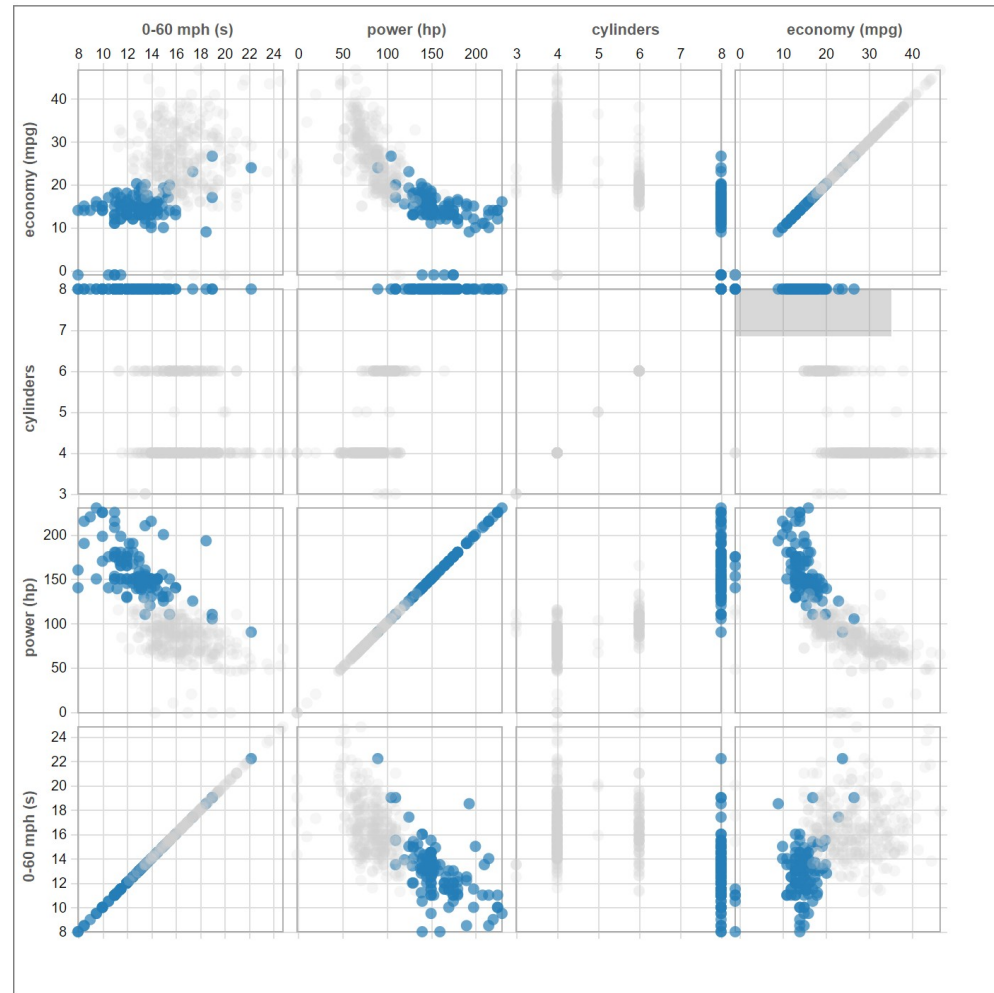
D3 Brush Object

```
var brush = d3.brush(); // create a new 2-D D3 brush object (use d3.brushX() or d3.brushY()
for 1-D brushes)

brush.extent([[topLeftX, topLeftY], [bottomRightX, bottomRightY]]); // define the
boundaries of the brushable area

brush.on('start', handleBrushStart); // add an event listener/callback for when the user
starts a new brush selection

brush.on('brush', handleBrushMove); // add an event listener/callback for when the user
drags the brush selection around

brush.on('end', handleBrushEnd); // add an event listener/callback for when the user
releases the mouse
```

D3 Brush Object

- To use the brush object, attach it to each chart and use `.call()` to activate it as you would with a chart axis.
- Example: `chartEnter.append('g').attr('class', 'brush').call(brush);`
- It's a good idea to make the brush appear *behind* the data; to do this, append the brush `<g>` *before* appending any of the data elements to the chart.

Brush Start Event

- Here, we can save a record of which chart we're in (requires a key/identifier value bound to each chart), as well as choose a scale domain/range for each axis, since the scales may vary between charts.
- `brush.move(brushGroup, null)` can be used to clear the brush's previous selection, where `brushGroup` is the <g> element that the brush is attached to.

Brush Move Event

- Use `d3.event.selection` to retrieve the boundaries of the brush's selection. (Example: `var [[left, top], [right, bottom]] = d3.event.selection;`)
  - JavaScript's array destructuring syntax above makes it easy to unpack the
- Given a scaled point `(x, y)` on our chart, we can see if it lies within the selection by checking whether `left <= x && x <= right && top <= y && y <= bottom`.

Brush End Event

- Clean up the brush selection by reverting the changes made during the move event (e.g., un-hide all data points that were hidden) and clearing the variable keeping track of the selected chart.

Bonus Content: D3 Tooltips

- The d3-tip library can be used to add a tooltip <div> that appears when the user hovers over an SVG element.

```
var toolTip = d3.tip().attr('class', 'd3-tip'); // instantiate a tooltip
object

toolTip.offset([offsetFromTop, offsetFromLeft]) // offset a tooltip
relative to its calculated position

toolTip.html(function(d) { ... }) // return an HTML string inside the
function to define the content that appears in the tooltip
```

Bonus Content: D3 Tooltips

- Use `svg.call(toolTip)` to activate the tooltip. Note that it is attached to the entire SVG canvas, rather than each individual chart <g> as with the brushes!
- Make the tooltip appear over your data-bound elements with `.on('mouseover', toolTip.show)` and disappear with `.on('mouseout', toolTip.hide)`.
- Example: `dotsEnter.on('mouseover', toolTip.show).on('mouseout', toolTip.hide);`

Bonus Content: D3 Tooltips

● The d3-tip library that I used is defined in a local .js file. To use it, include it in the same folder as your `index.html` and `main.js` files, and add <script src="./d3-tip.js"></script> to your `index.html` file.
● The `style.css` file I used in my example also contains some tooltip styling!

References

- I based my demo code off of a combination of examples from CS 4460 and CS 7450.
- I created my Nintendo Gamecube game dataset by manually taking data from gamevaluenow.com (price/year data), gamefaqs.com (rating/difficulty/length/genre data), and vgchartz.com (sales data).