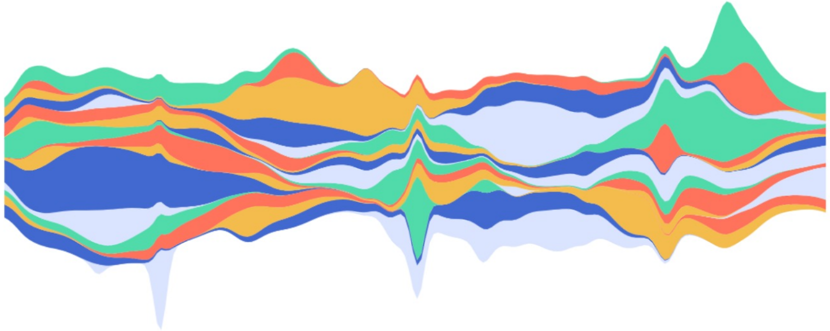
D3 by Observable

Introduction to D3

Session 2

[Observable notebook]

[YouTube video]



These slides are provided with the course “Introduction to D3” by Paul Buffa – Head of Product Education at Observable

Agenda

1. What makes a Scatterplot?
2. Scales
3. The Canvas
4. Data joins
5. Our 1st Scatterplot

What makes a Scatterplot?

What makes a Scatterplot?

As we covered in session 1, working with D3 means we have to build everything we see in our visualization. Let’s look at a Scatterplot then and figure out all the components we need to build.

Activity 1

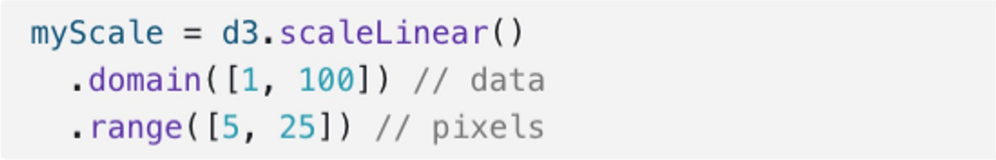
Scales

What is a scale?

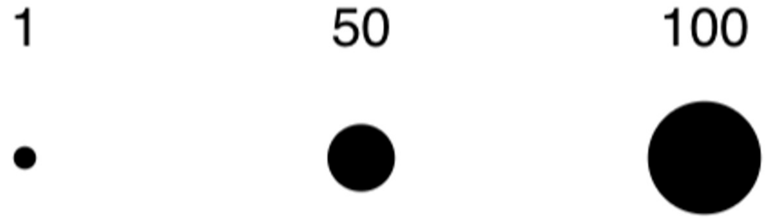
In D3, a scale is a function that maps an abstract dimension of data to a visual representation.

What is a scale?

Here we have a linear scale that takes data (1, 50, 100) and uses it to change the radius of circles.

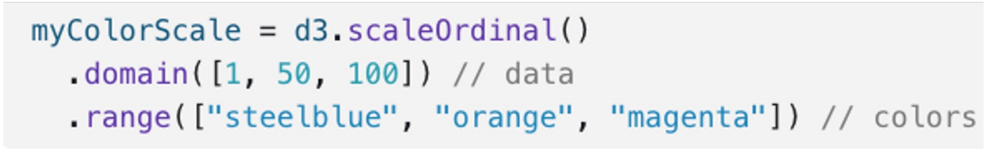


```
myScale = d3.scaleLinear()
  .domain([1, 100]) // data
  .range([5, 25]) // pixels
```

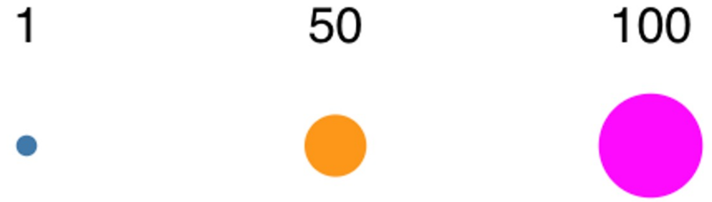


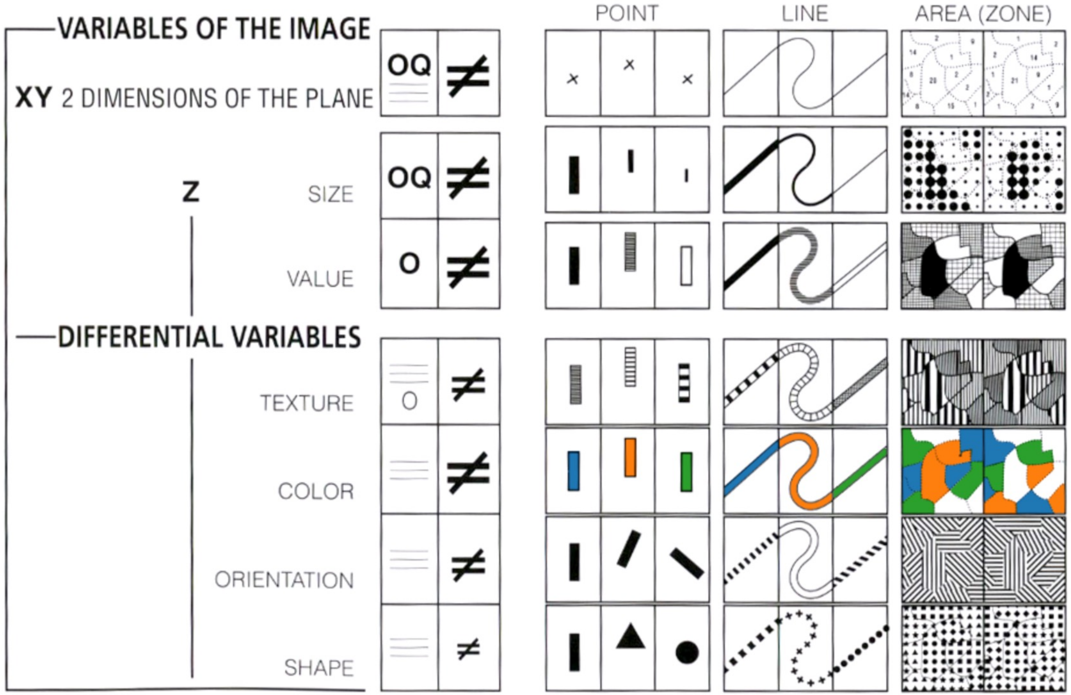
What is a scale?

This scale takes the data and sets colors to it. It maps each point 1:1. This is called an ordinal scale.



```
myColorScale = d3.scaleOrdinal()
  .domain([1, 50, 100]) // data
  .range(["steelblue", "orange", "magenta"]) // colors
```





From *Semiology of Graphics*, colorized by the author.

Activity 2 & 3

The Canvas

What is the canvas?

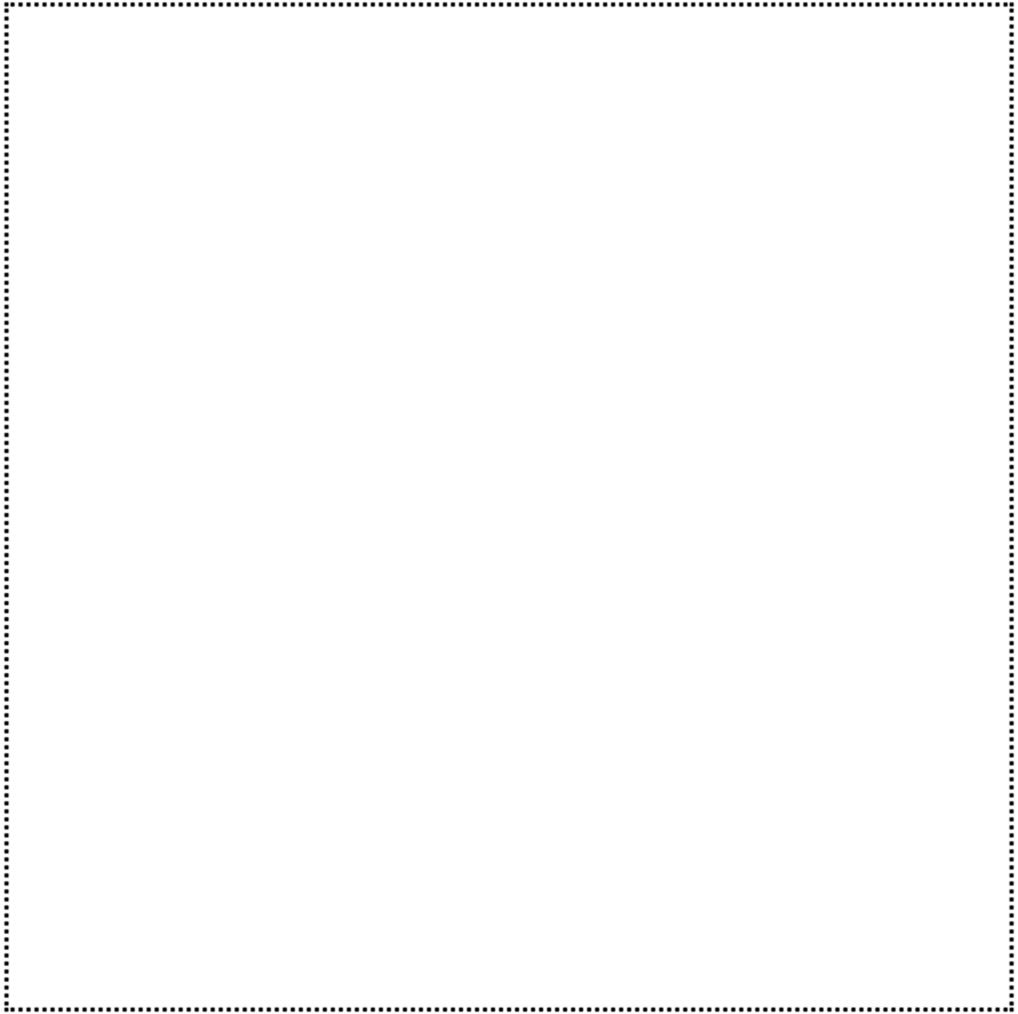
Not to be confused with HTML canvas, which we will not be covering in this course, generally when we're referring to a canvas, we're talking about the **medium which we render our visualization.**

Our canvas

Here is our blank canvas. Looks great, right?

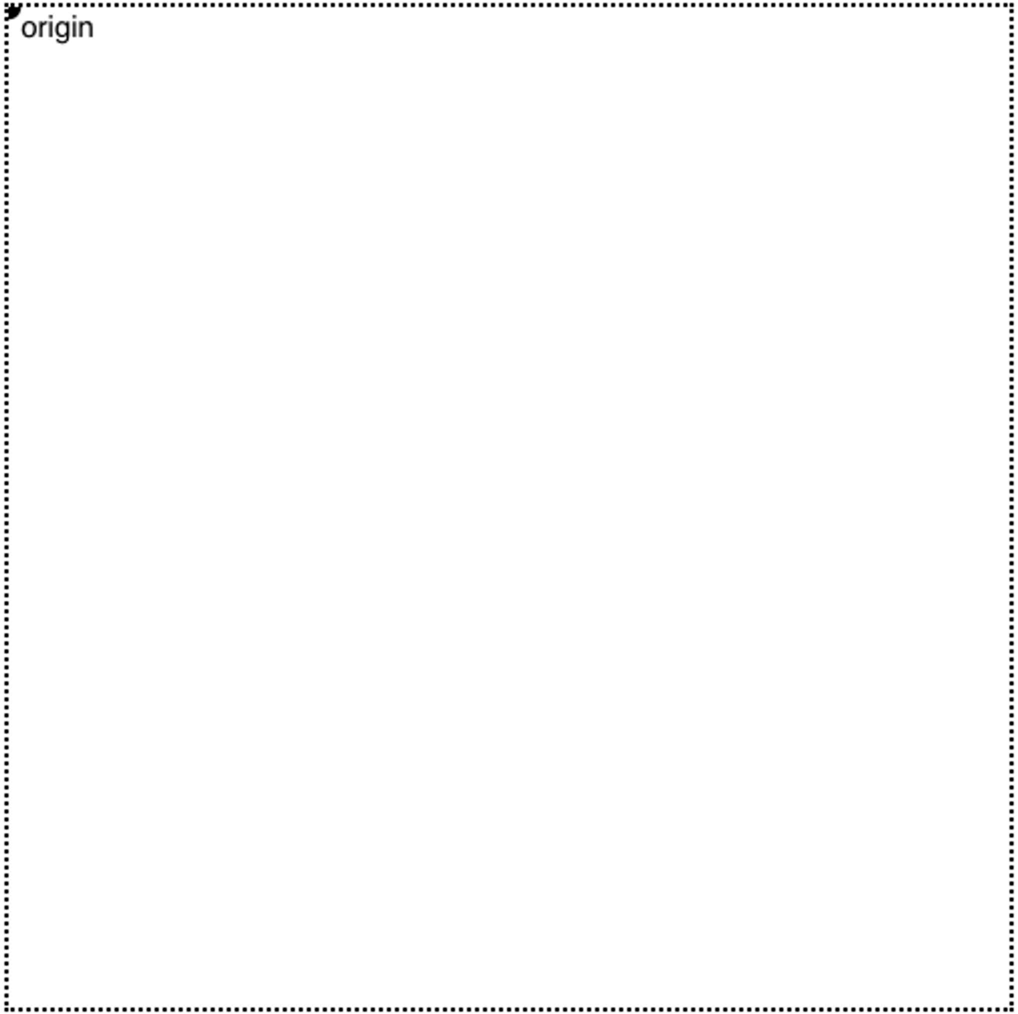
Our canvas

We need to add a border so we can see it.



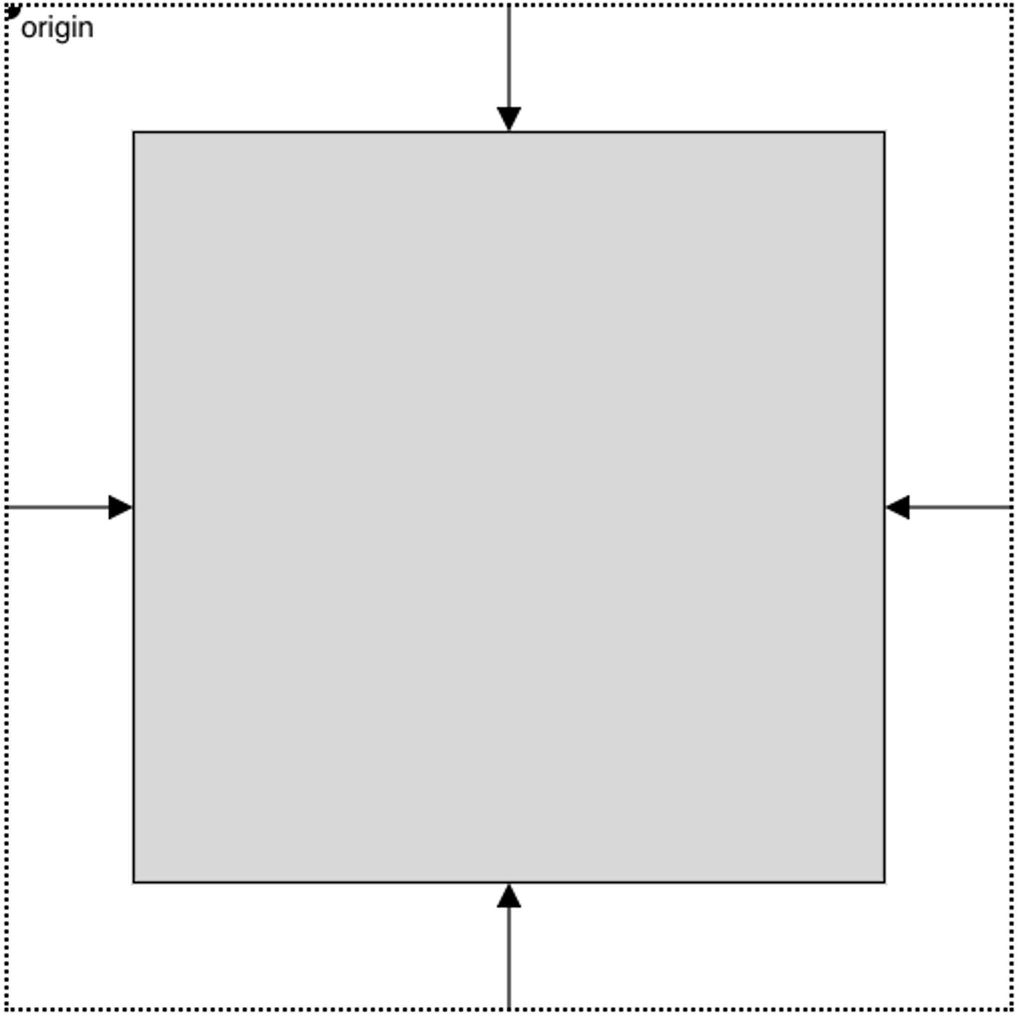
Our canvas

If we add a circle without giving coordinates, it places it at the origin.



Our canvas

When creating a chart, we often want some margins so our axes don’t get cut off, like the circle did.



Activity 4

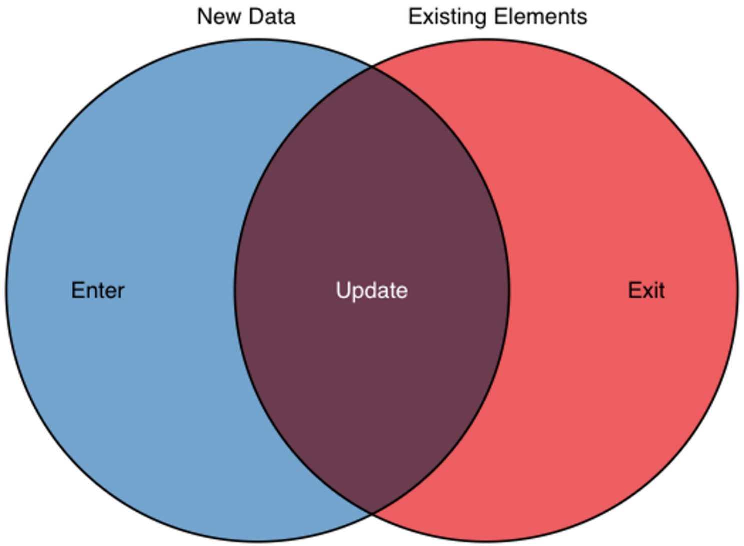
Data Joins

What is a data join?

The data join is arguably the most confusing part of learning D3, which is how we create objects, i.e. circles, rectangles, lines, for the number of datapoints in our dataset.

The data join

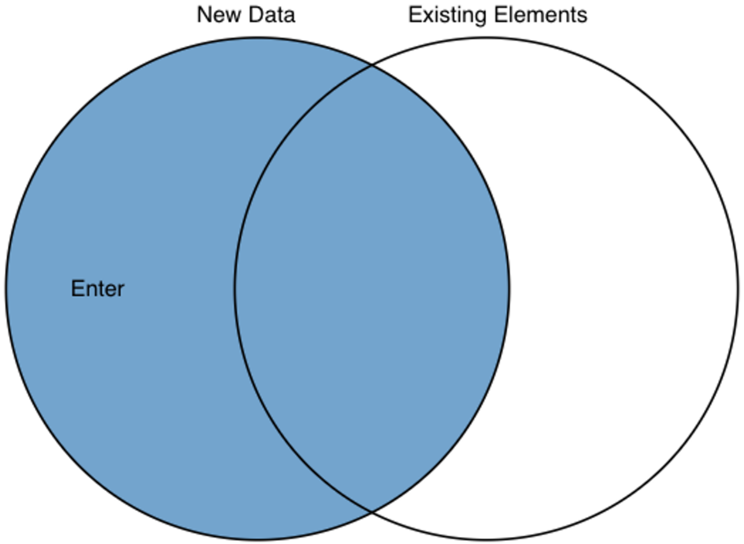
To join data with SVG elements, we compare the two sets, and determine whether we need to enter, update or exit, i.e. the general update pattern.



The three selections computed by the *selection*.data.

The data join

When we first create our visualization there is no existing elements, so typically start by entering in new elements to match our data.



The three selections computed by the *selection*.data.

What is a data join?

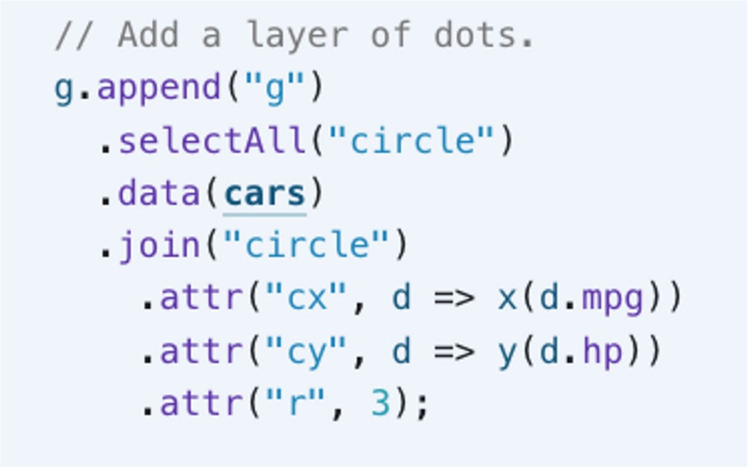
But how do we know there is no existing elements?

What is a data join?

But how do we know there is no existing elements?
We start our join by trying to select all the elements that we want, then comparing to data, and evaluate with a join.

The data join

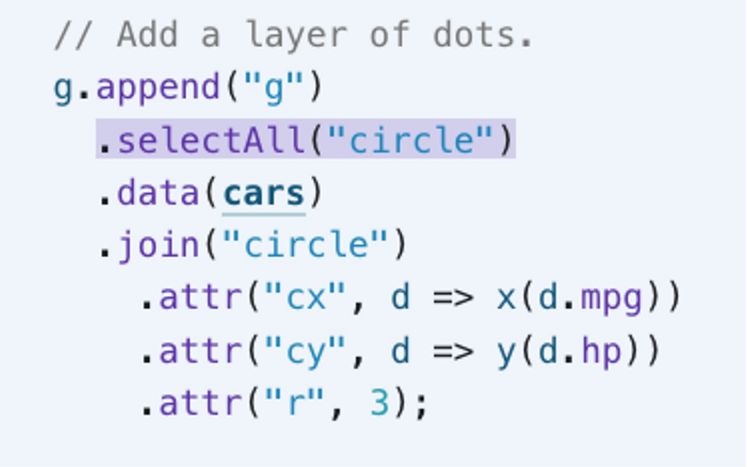
Here is an example of a data join where we create circles based on the number of cars in our cars dataset.



```
// Add a layer of dots.
g.append("g")
  .selectAll("circle")
  .data(cars)
  .join("circle")
    .attr("cx", d => x(d.mpg))
    .attr("cy", d => y(d.hp))
    .attr("r", 3);
```

The data join

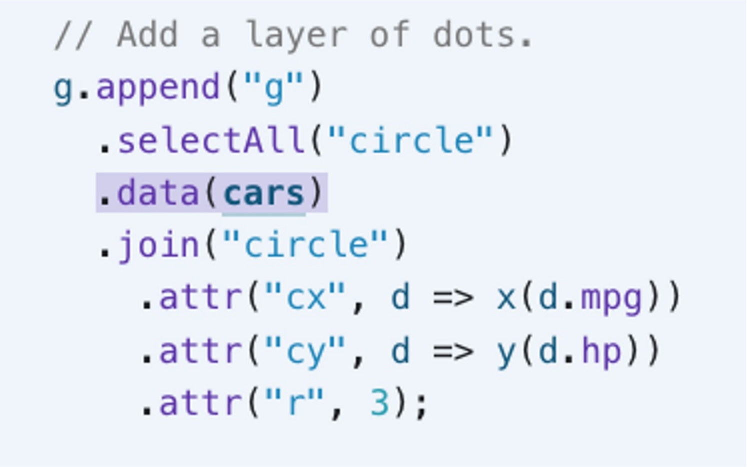
Notice how we start by selecting all the circles that don’t exist. This is how we know in our Venn diagram whether the elements exist.



```
// Add a layer of dots.
g.append("g")
  .selectAll("circle")
  .data(cars)
  .join("circle")
    .attr("cx", d => x(d.mpg))
    .attr("cy", d => y(d.hp))
    .attr("r", 3);
```

The data join

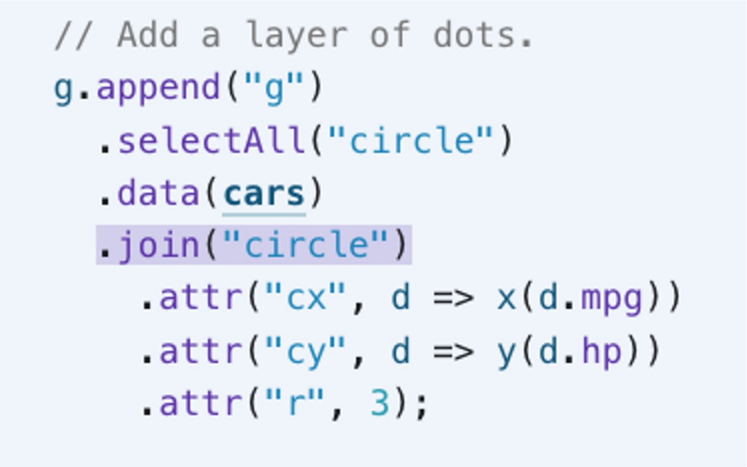
The second part of our Venn diagram is to compare to our data, so we use .data() and enter our cars dataset.



```
// Add a layer of dots.
g.append("g")
  .selectAll("circle")
  .data(cars)
  .join("circle")
    .attr("cx", d => x(d.mpg))
    .attr("cy", d => y(d.hp))
    .attr("r", 3);
```

The data join

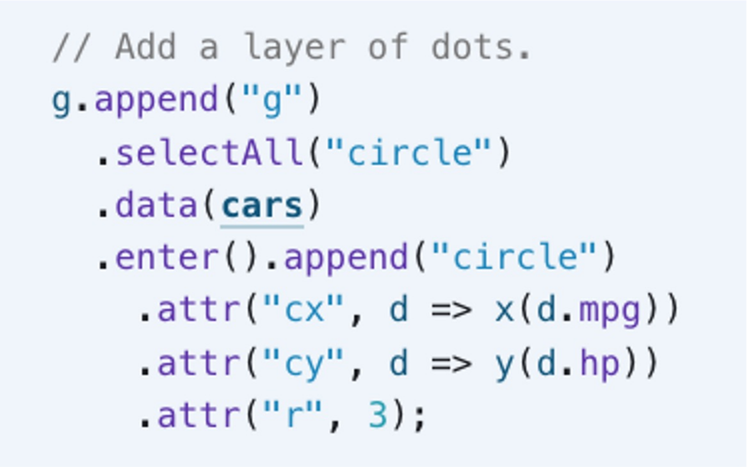
Finally, we use the join method to compare our selection to the dataset and determine whether we need to enter, update or exit the number of items.



```
// Add a layer of dots.
g.append("g")
  .selectAll("circle")
  .data(cars)
  .join("circle")
    .attr("cx", d => x(d.mpg))
    .attr("cy", d => y(d.hp))
    .attr("r", 3);
```

The data join

The join method was introduced in a later version of D3. Earlier versions, you had to know whether you wanted to enter, update or exit.



```
// Add a layer of dots.
g.append("g")
  .selectAll("circle")
  .data(cars)
  .enter().append("circle")
    .attr("cx", d => x(d.mpg))
    .attr("cy", d => y(d.hp))
    .attr("r", 3);
```

Our 1st Scatterplot

Activity 5

Session 2 is a wrap!

Observable