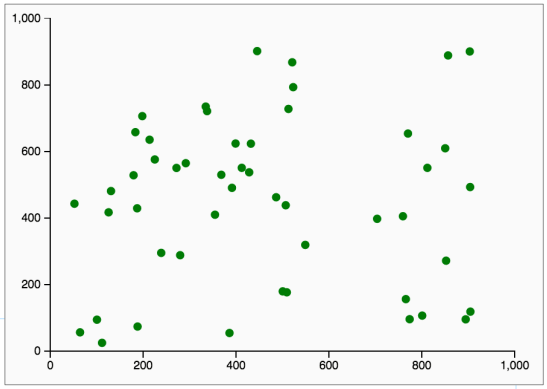
ANIMATING WITH TRANSITIONS IN D3

ADDING SIMPLE TRANSITIONS

D3 makes animating transitions between chart states embarrassingly easy

- Whenever we make a change to the attributes or style of DOM elements we can ask D3 to animate the transition to that new value rather than changing instantly



6_1_1_D3_Simple_Transitions.html

transitions.duration

duration is a D3 method that sets the duration of a transition

The duration function takes one parameter

- duration — The duration for the transition in milliseconds. If no duration is specified the default duration of 250ms is used.

Scales API reference: https://github.com/d3/d3-3.x-api-reference/blob/master/API-Reference.md#d3scale-scales

selection.transition

transition is a D3 method that starts a transition for a selection

The transition function takes one optional parameter

- name — A string name of the transition that allows multiple transitions to be controlled

transition API reference: https://github.com/d3/d3-3.x-api-reference/blob/master/Selections.md#transition


```
/******** HANDLE UPDATE SELECTION ************/
circles
   .transition()
   .duration(500)
   .attr("cx", function(d) {
                  return xScale(d[0]);
               })
   .attr("cy", function(d) {
                  return yScale(d[1]);
                })
   .attr("r", 4)
   .style("fill", "green");
```

DIFFERENT TYPES OF TRANSITIONS - EASING

We can control the way that the transition happens using easing

There are a number of different types of easing

- cubic-in-out accelerate up and down
- linear no acceleration
- elastic simulates an elastic band
- back simulates backing into a parking space
- bounce simulates a bouncy collision

Great easing reference: http://easings.net/

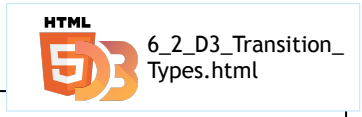
transition.ease

ease is a D3 method that sets the style of transition that will be applied to a selection

The ease function takes one optional parameter

- name A string name of the transition that allows multiple transitions to be controlled
- params Parameters for the easing function

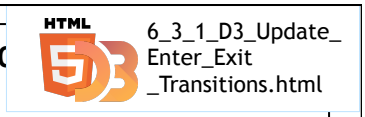
transition API reference: https://github.com/d3/d3-3.x-api-reference/blob/master/Selections.md#transition




```
/******** HANDLE UPDATE SELECTION ************/
circles
   .transition()
   .duration(500)
   .easing("bounce")
   .attr("cx", function(d) {
                  return xScale(d[0]);
              })
   .attr("cy", function(d) {
                  return yScale(d[1]);
               })
   .attr("r", 4)
   .style("fill", "green");
```

ADDING TRANSITIONS FOR UPDATE, ENTER, AND EXIT SELECTIONS

If we have data that is changing we can add different transitions for update, enter, and exit selections



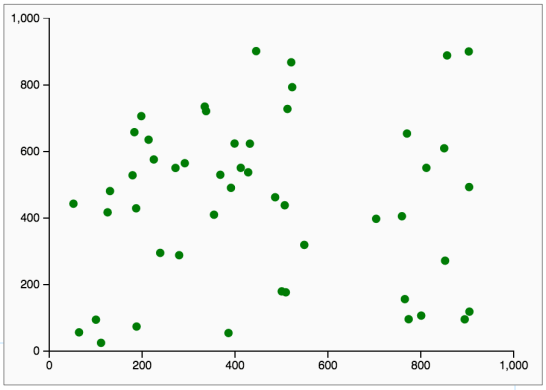

```
/******** HANDLE UPDATE SELECTIO
circles
   .transition()
   .duration(2000)
   .style("fill", "green");

/******** HANDLE ENTER SELECTION ************/
circles.enter()
   .append("circle")
   .transition()
   .duration(2000)
   .style("fill", "Blue");

/******** HANDLE EXIT SELECTION ************/
circles.exit()
   .remove();
```

We need to watch out for a few things

- Enter selections need a starting point from which to start transitions - otherwise starting location is (0, 0)
- Exit selections need something to happen before removal



6_3_2_D3_Update_Enter_Exit_Transitions.html

SCHEDULING TRANSITIONS

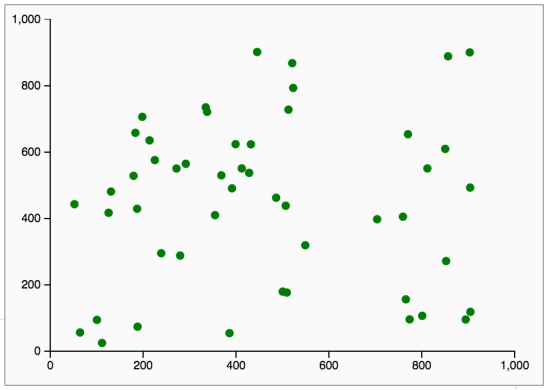
transition.delay

We can add delays to transitions to control the timing with which different things happen

delay is a D3 method that adds a delay before a transition starts

The delay function takes one optional parameter

- delay — The delay in milliseconds - note either a constant delay or a delay calculated using a function



6_4_D3_Scheduling_Transitions.html

INTERACTIONS IN D3

ADDING INTERACTIONS IN D3

We add interactions in D3 by adding event listeners to elements and the defining callback functions that handle these events

We can use any events supported by the browser, for example:

- click
- dblclick
- mousedown
- mouseenter
- mouseleave
- mouseover
- mouseup
- keydown
- keyup
- keypress

Full list of browser events from Mozilla: https://developer.mozilla.org/en-US/docs/Web/Events#Standard_events

selection.on

on is a D3 method that adds an event listener to an element

The on function takes two parameters

- type — The type of event to listen for (e.g. click, mousedown, …)
- listener — A function defined to handle the event

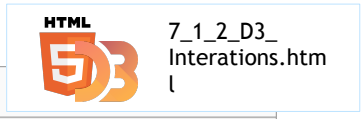
this

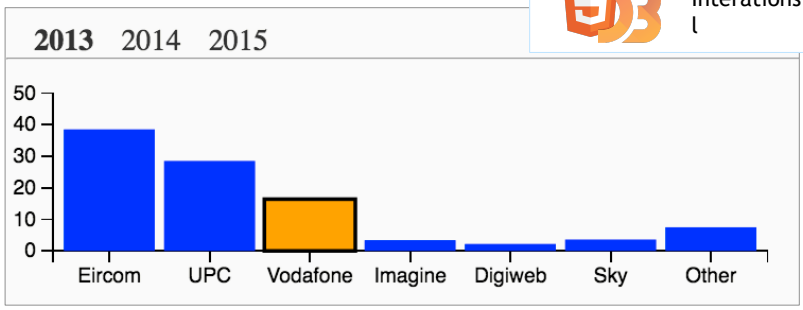
In the event handler code we can refer to the object that caused the event to occur using the this keyword


```
d3.selectAll(".button")
   .on("click", function(d) {
      // Update the display year
      display_year = d3.select(this).text();

      // Update the visualisation
      generateVis();
});
```

We can add event listeners to any element we draw in a D3 visualisation





ADDING TOOLTIPS IN D3

The simplest way to add tooltips is just to use HTML tooltips

Append a title element to any SVG element to add a tooltip

- Set the tooltip text using the text function

7_2_D3_ Tooltips.html

TIMED EVENTS

We can make nice animations using transitions and timed events in D3

There are two key methods required to implement timed events:

- setInterval Sets a callback function to be executed at a regular time interval
- clearInterval Stops a callback that has been set up using setInterval

setInterval

setInterval is a utility JavaScript method for calling a function at a regular interval

The setInterval method takes one parameter

- callback A callback function that will be repeatedly called
- milliseconds The frequency with which the callback function will be called

We can save the ID value returned by setInterval so that we can turn it off with clearInterval later

clearInterval

clearInterval is a utility JavaScript method for stopping a regular callback initialised by setInterval

The clearInterval method takes one parameter

- ID — The ID returned by setInterval identifying the callback we want to end

7_3_D3_Timed_Events.html