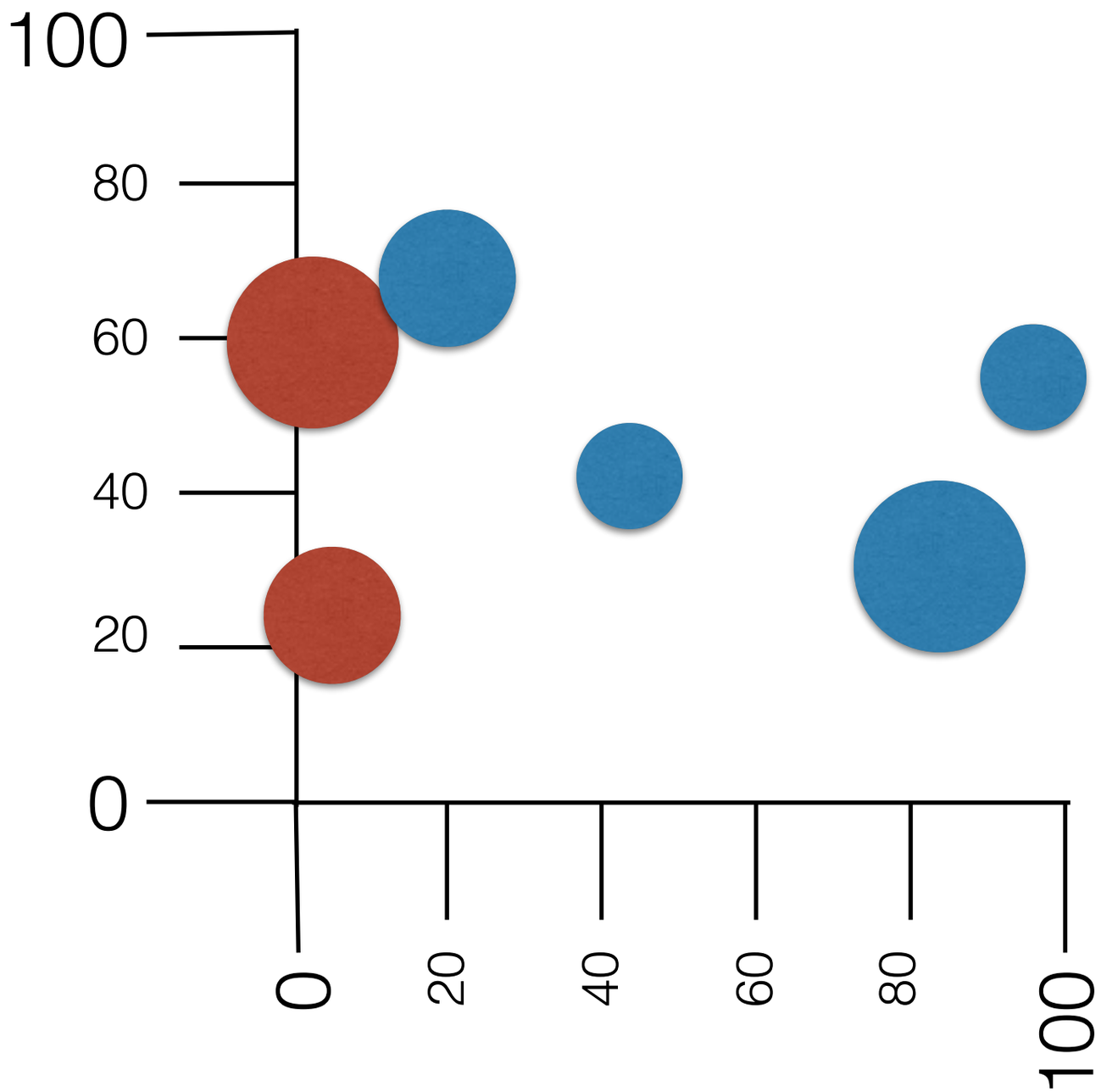
Enter, Update, Exit

Lab 5

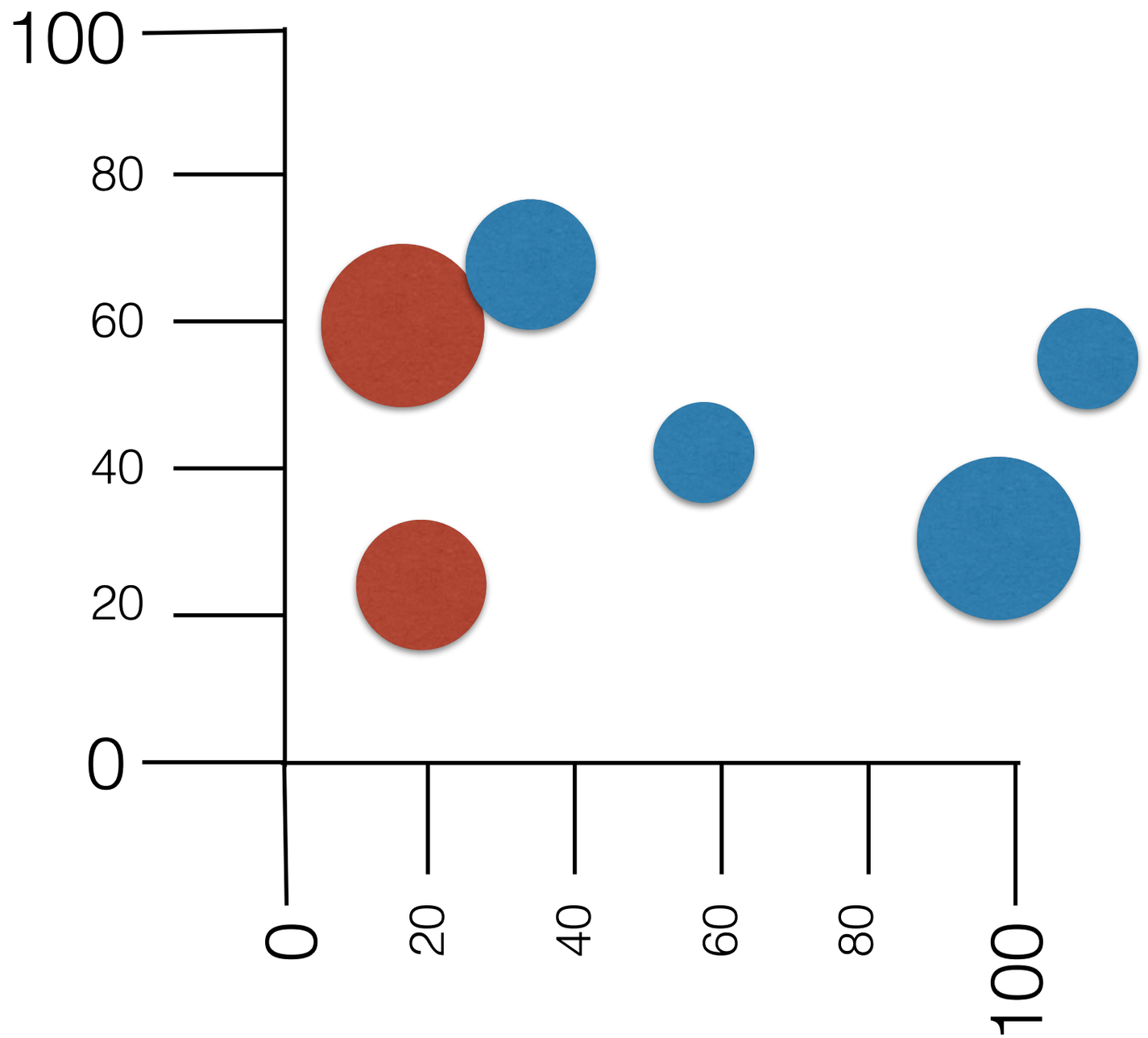
Feedback Lab 4

- Axis placement
- Axis drawing
- The ‘g’ element

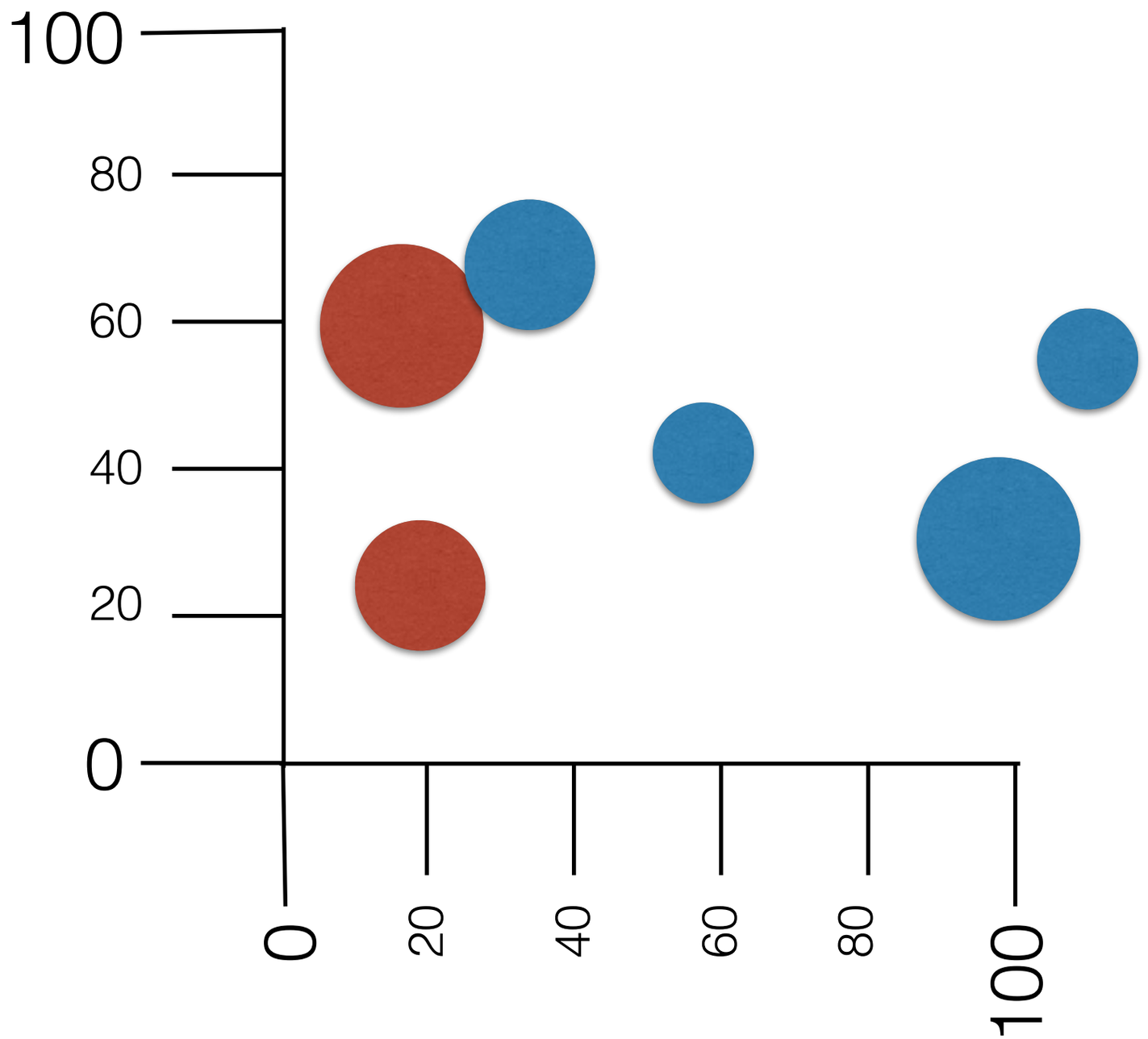
This happens…



Solution ?

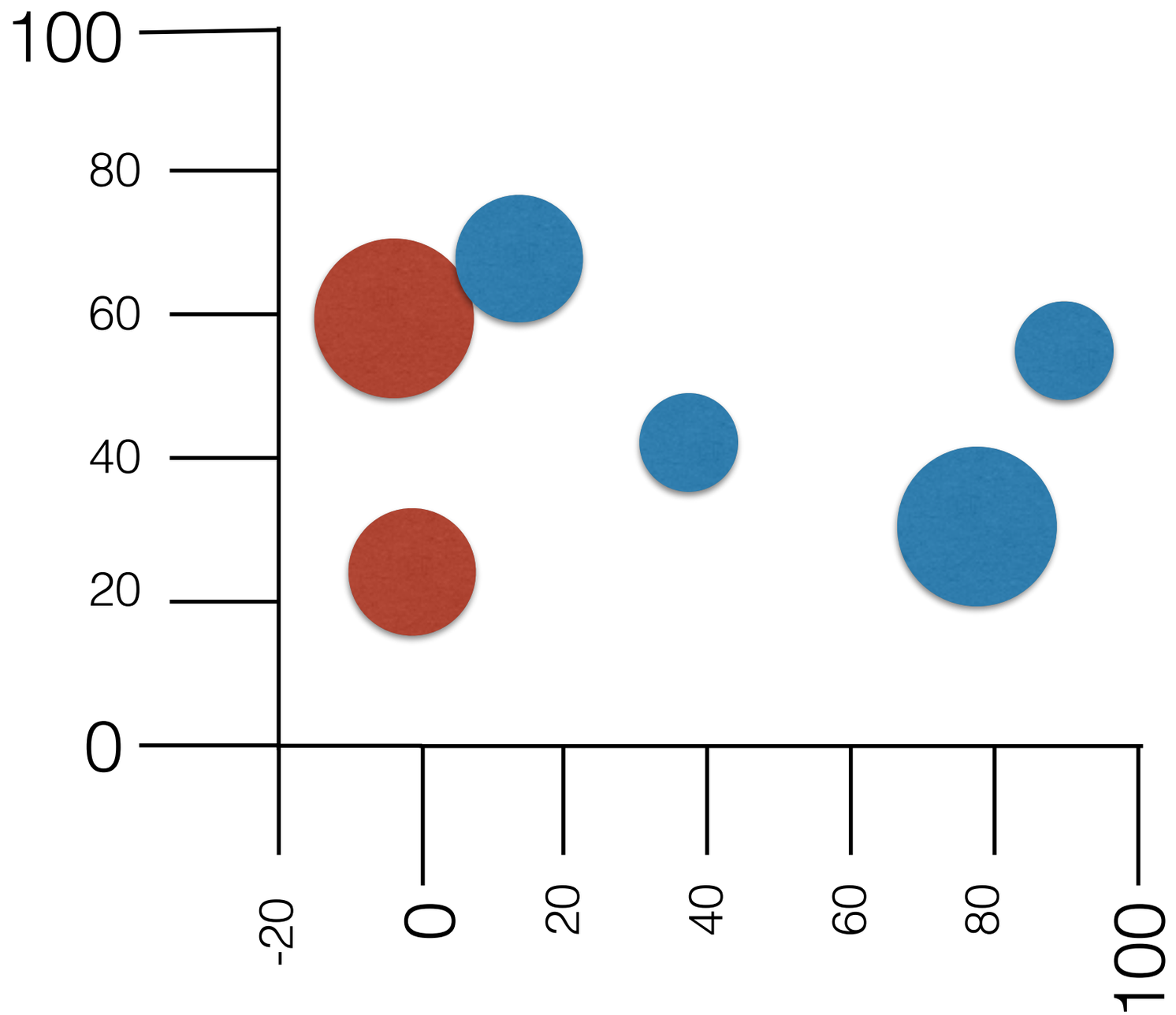


Solution ?

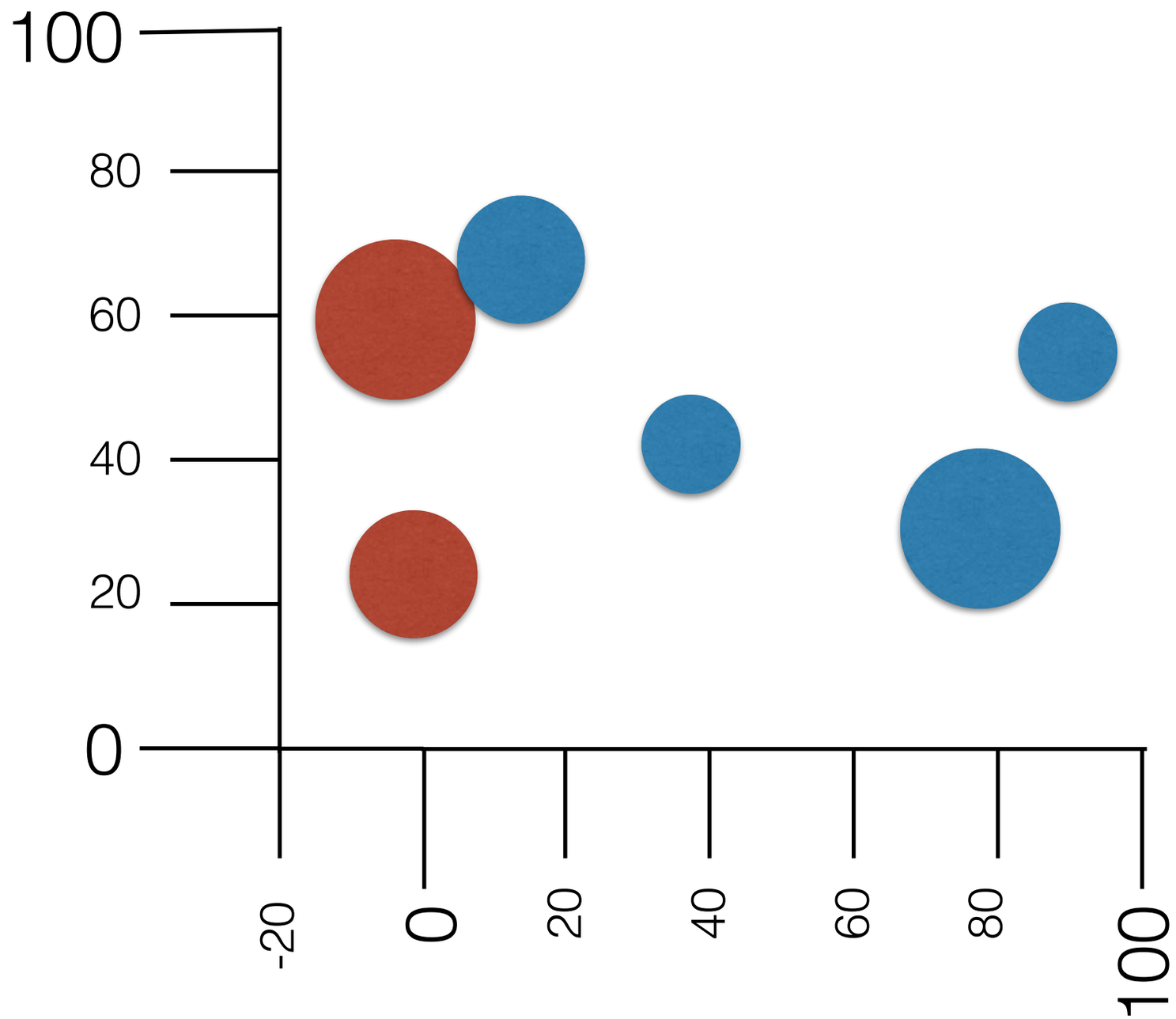


NO !!

Solution ?

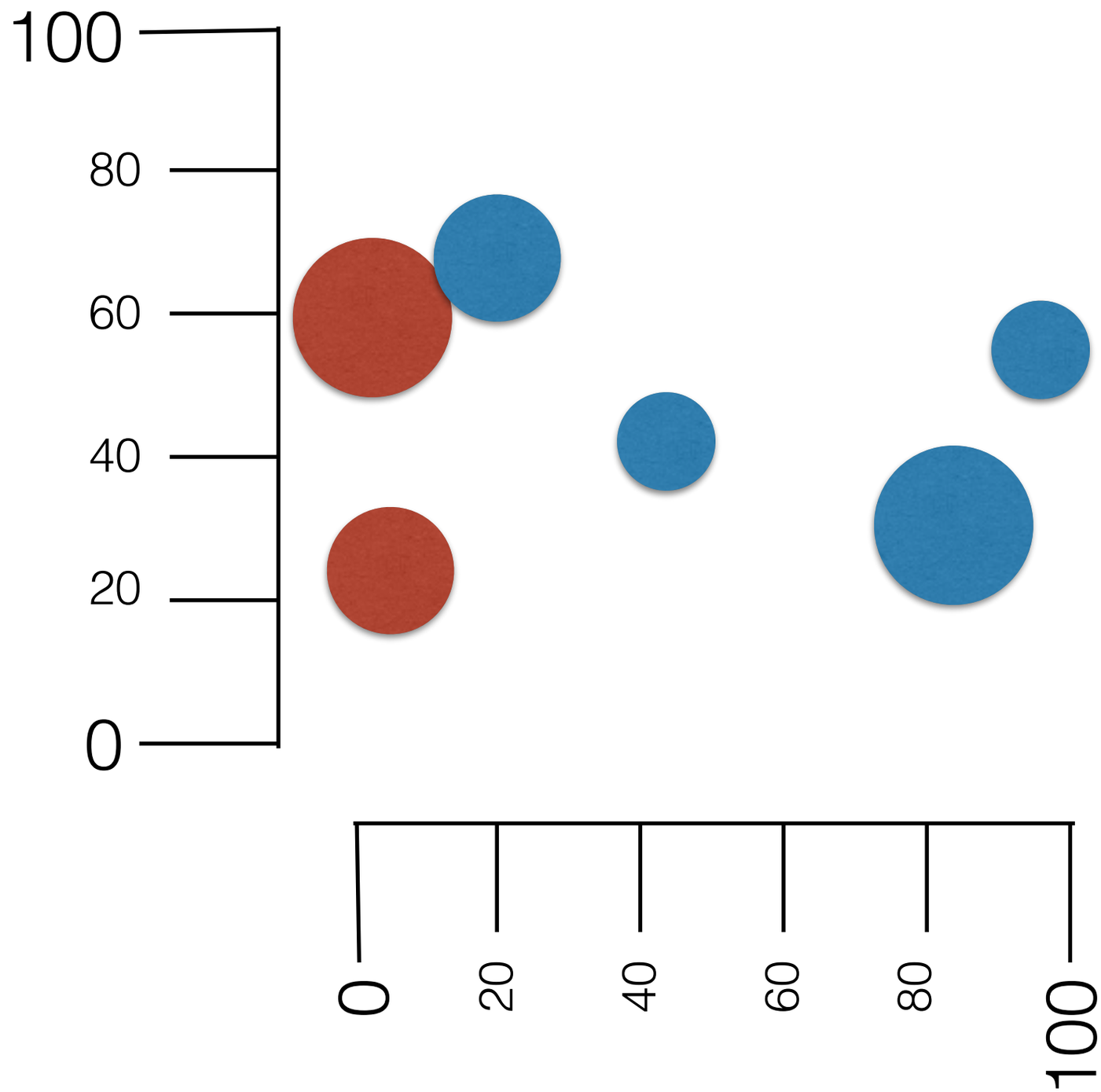


Solution ?

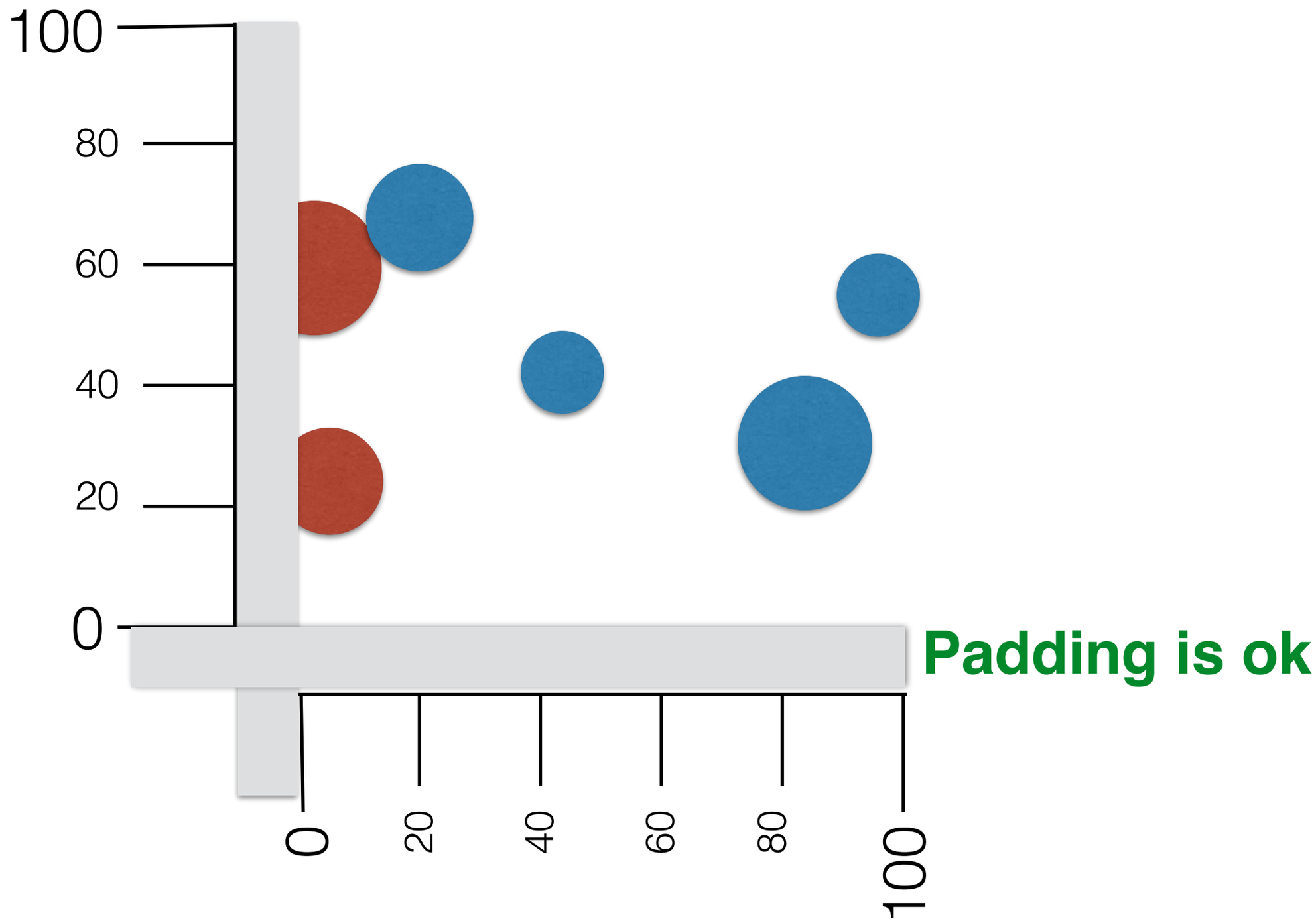


ok, BUT bad style

Solution ?

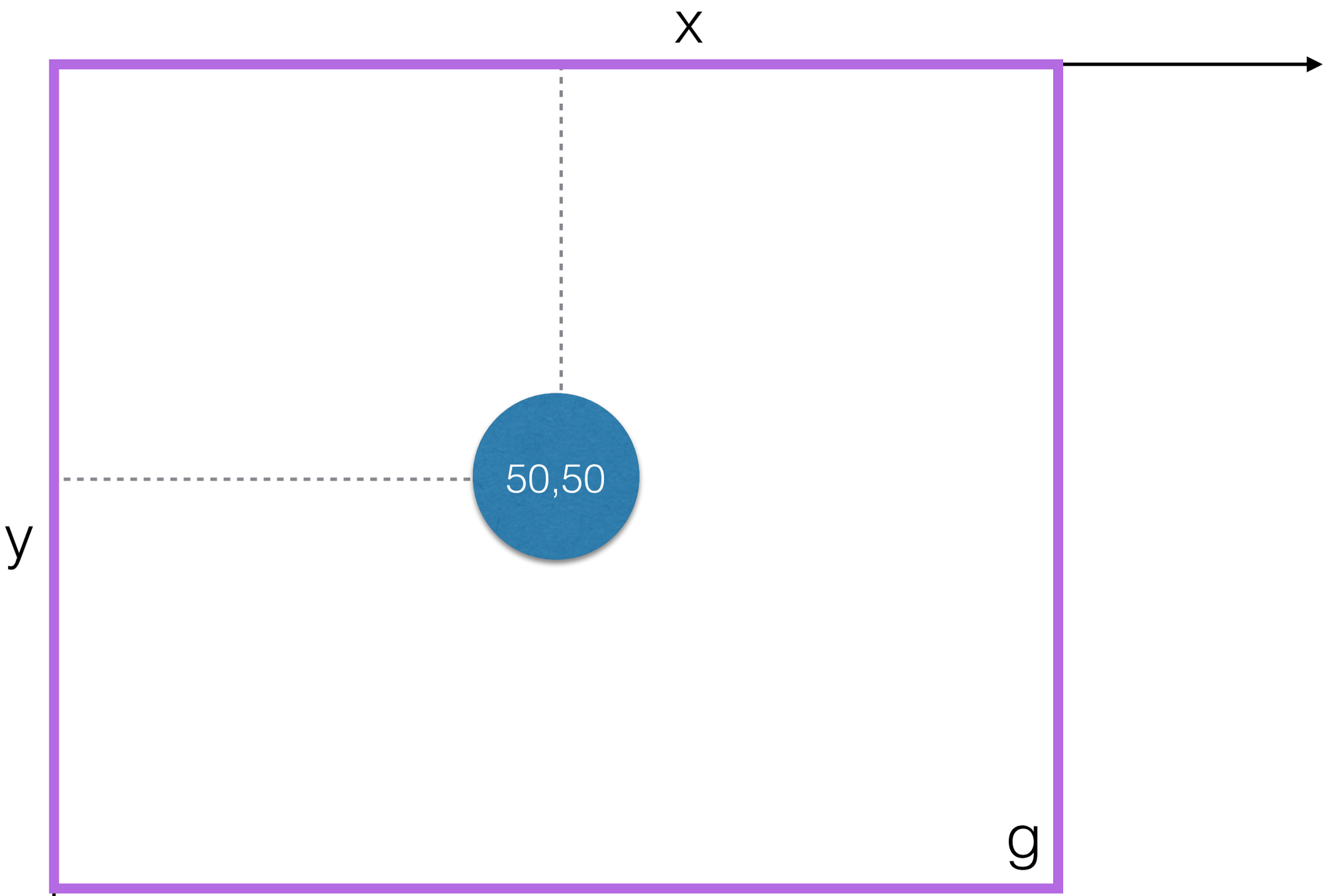


Solution ?

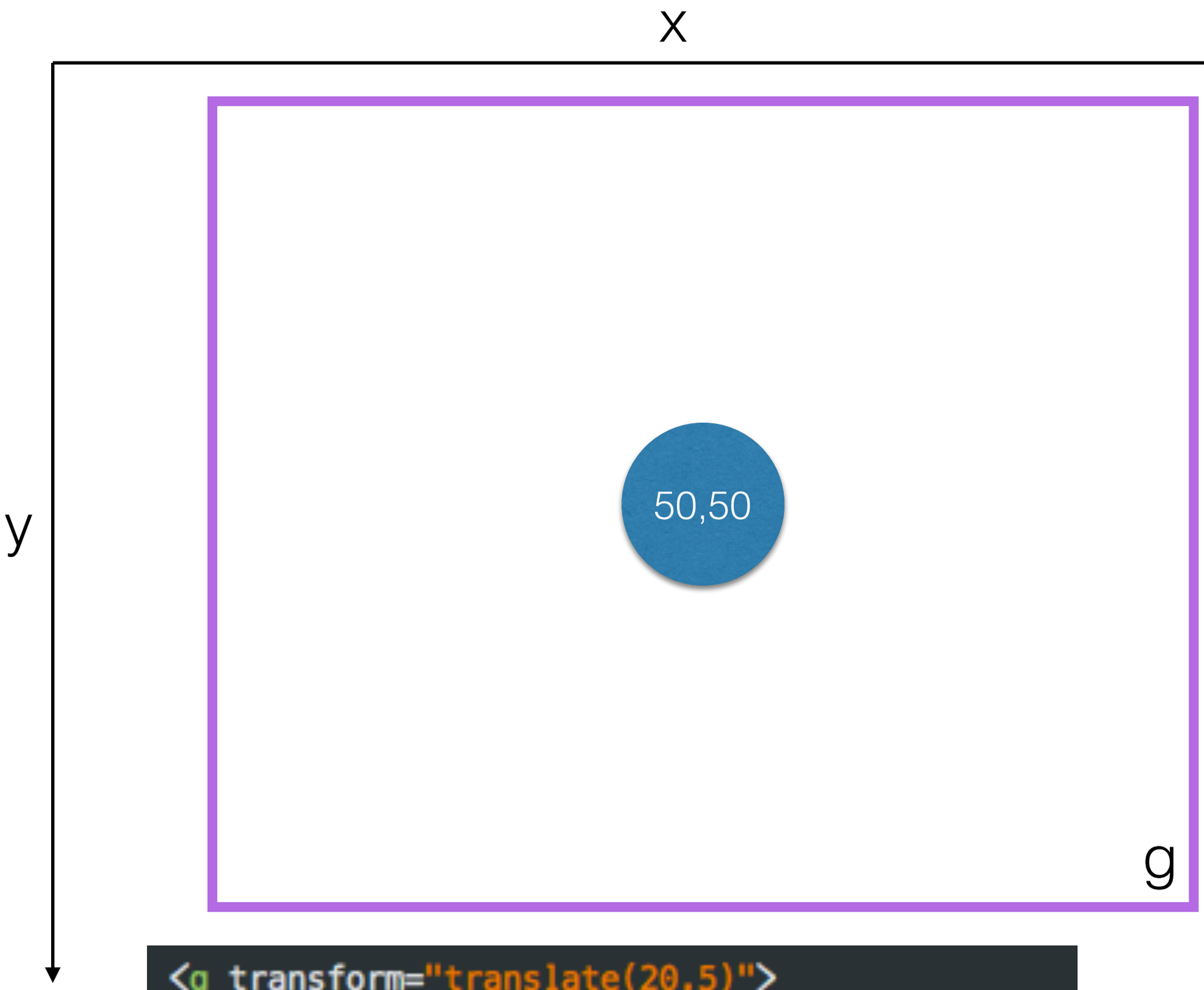


The ‘g’ element

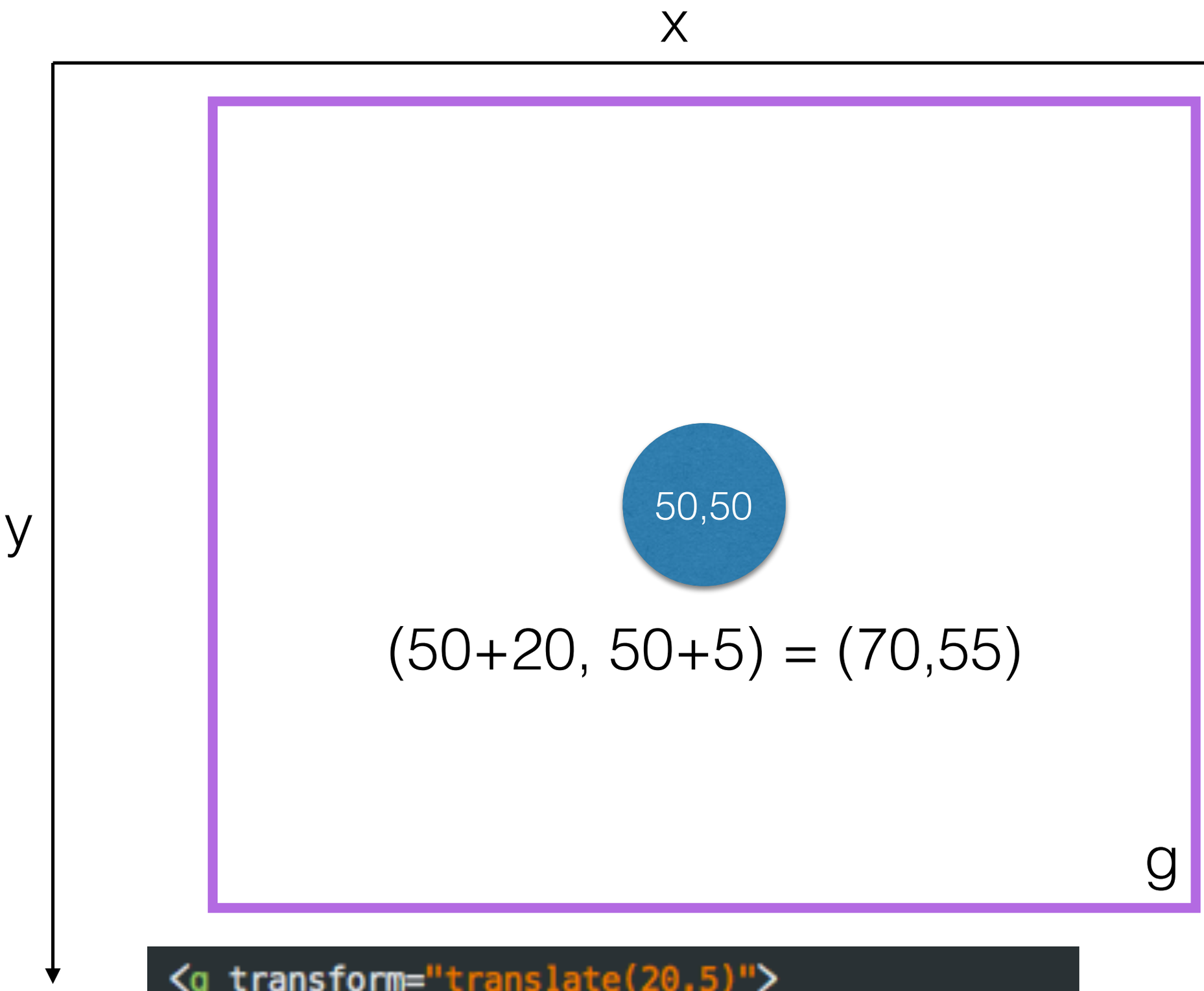
- groups visual elements
- starts a new coordinate system
- all transformations to the ‘g’ apply to all elements in the group
- think of grouping in Keynote or Powerpoint




```
<g>
    <circle r="10" cx="50" cy ="50"></circle>
</g>
```



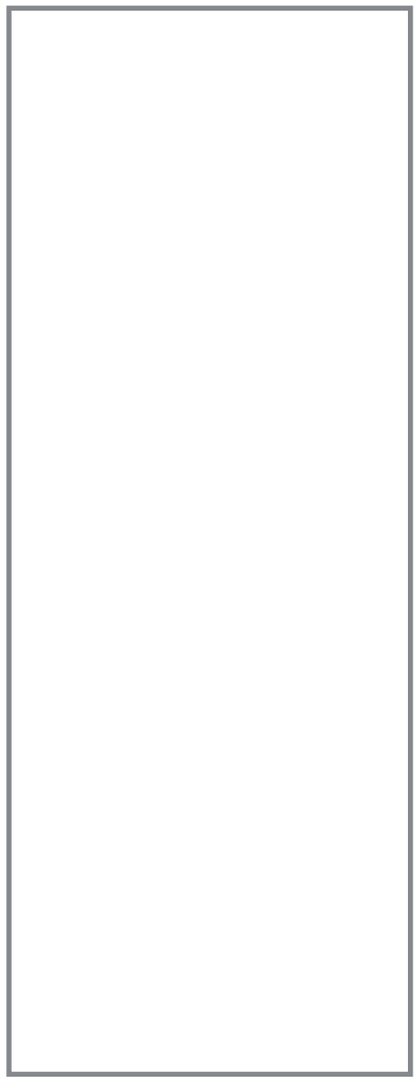

```
<g transform="translate(20,5)">
    <circle r="10" cx="50" cy ="50"></circle>
</g>
```




```
<g transform="translate(20,5)">
    <circle r="10" cx="50" cy ="50"></circle>
</g>
```

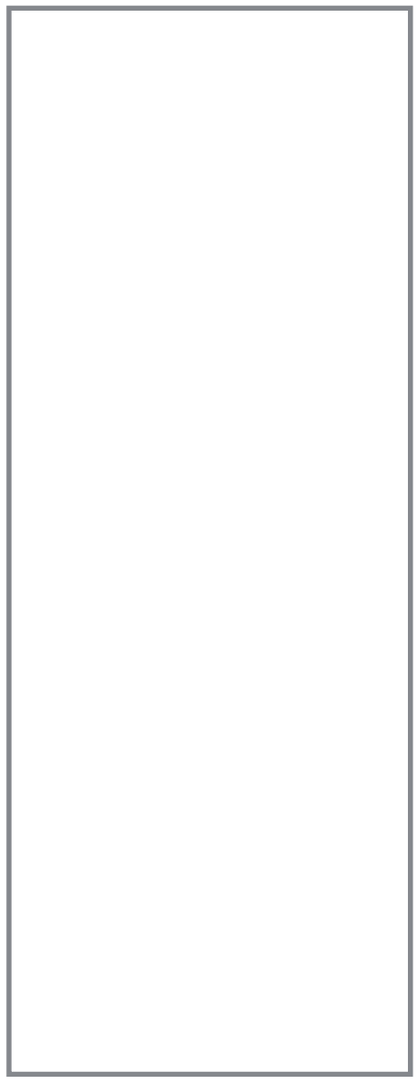
The ‘g’ element is a container for D3 to draw an axis into.

<g id=‘blob’></g>



The ‘g’ element is a container for D3 to draw an axis into.

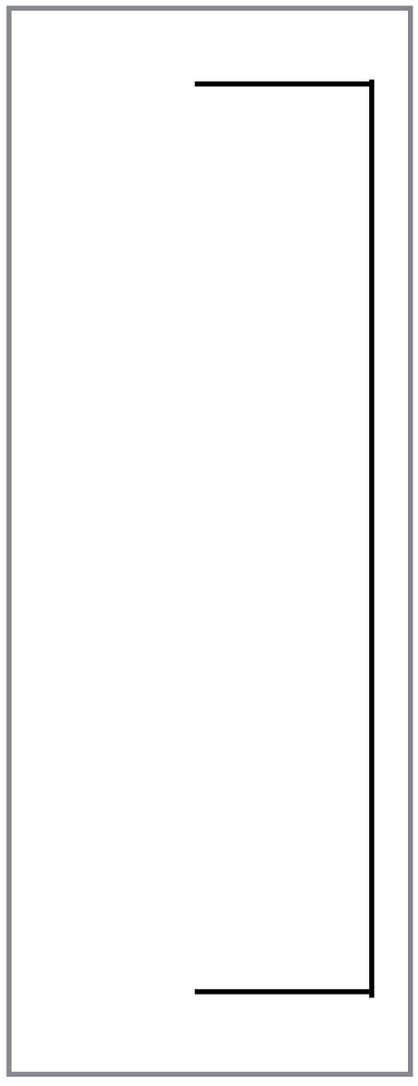
<g id=‘blob’></g>



```
d3.select('#blob').call(axis);
```

The ‘g’ element is a container for D3 to draw an axis into.

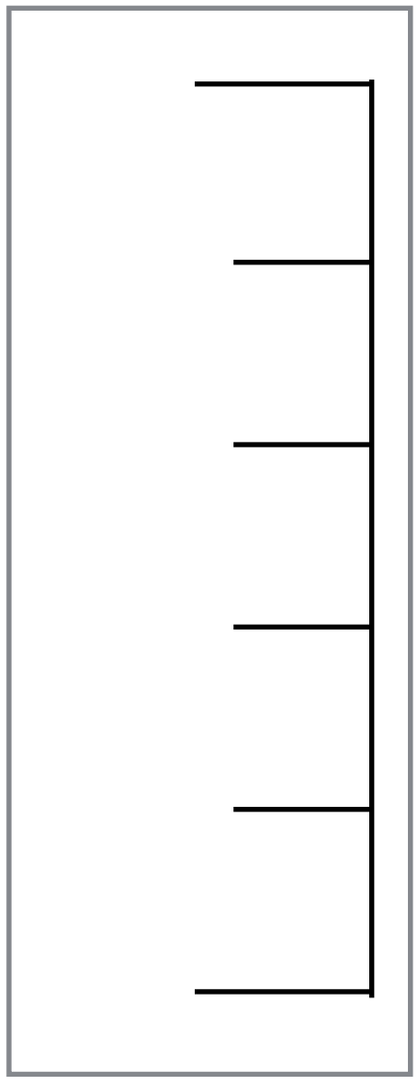
<g id=‘blob’></g>



```
d3.select('#blob').call(axis);
```

The ‘g’ element is a container for D3 to draw an axis into.

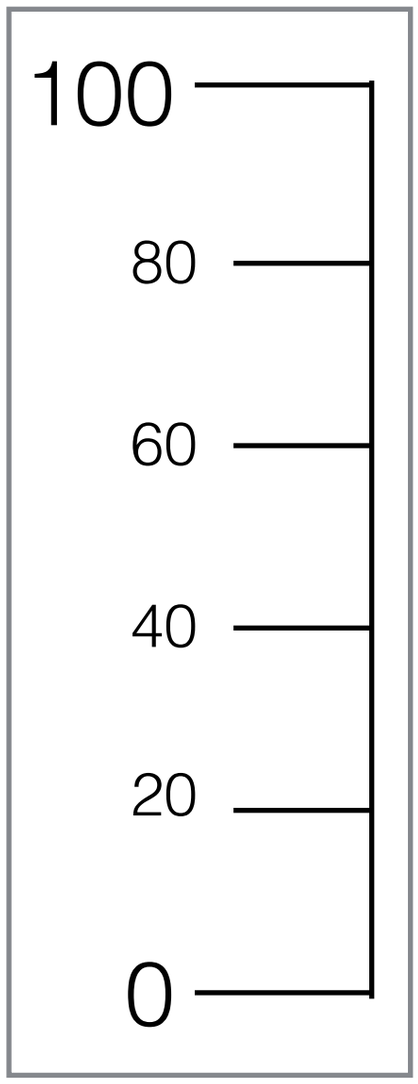
<g id=‘blob’></g>



```
d3.select('#blob').call(axis);
```

The ‘g’ element is a container for D3 to draw an axis into.

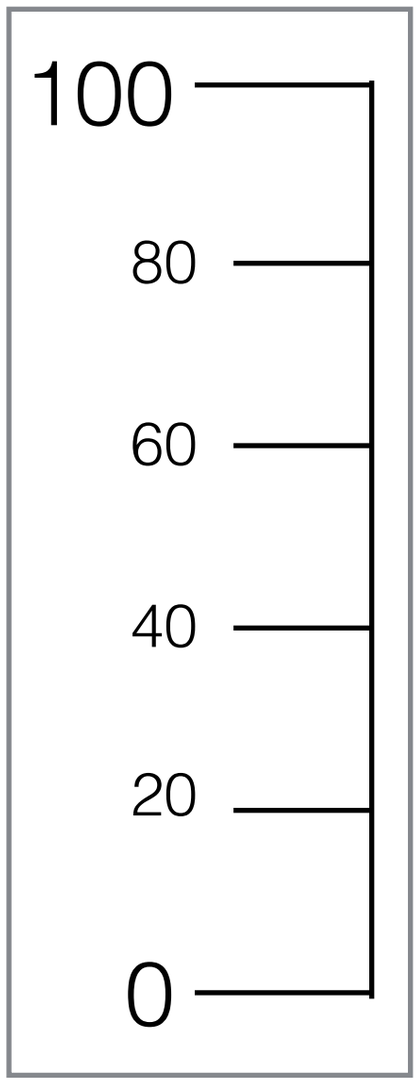
<g id=‘blob’></g>




```
d3.select('#blob').call(axis);
```

The ‘g’ element is a container for D3 to draw an axis into.

<g id=‘blob’></g>

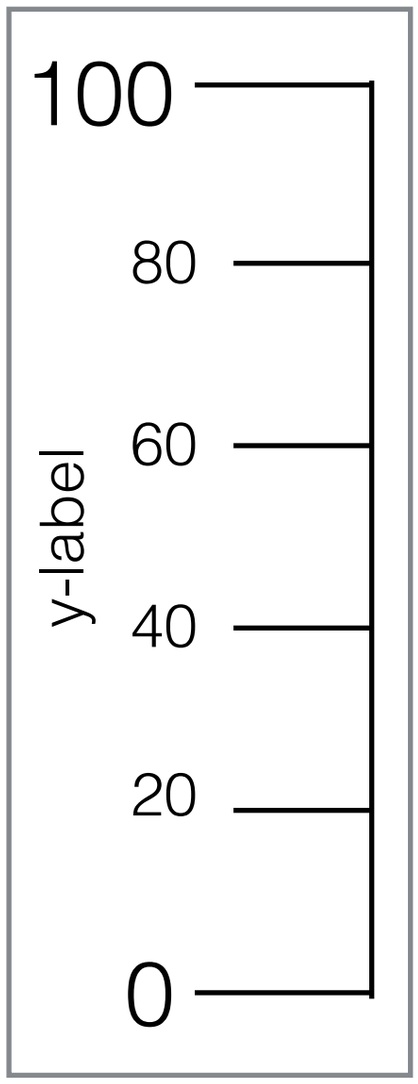



```
d3.select('#blob').call(axis);
```

```
d3.select(‘#blob’).append(‘text’)…
```

The ‘g’ element is a container for D3 to draw an axis into.

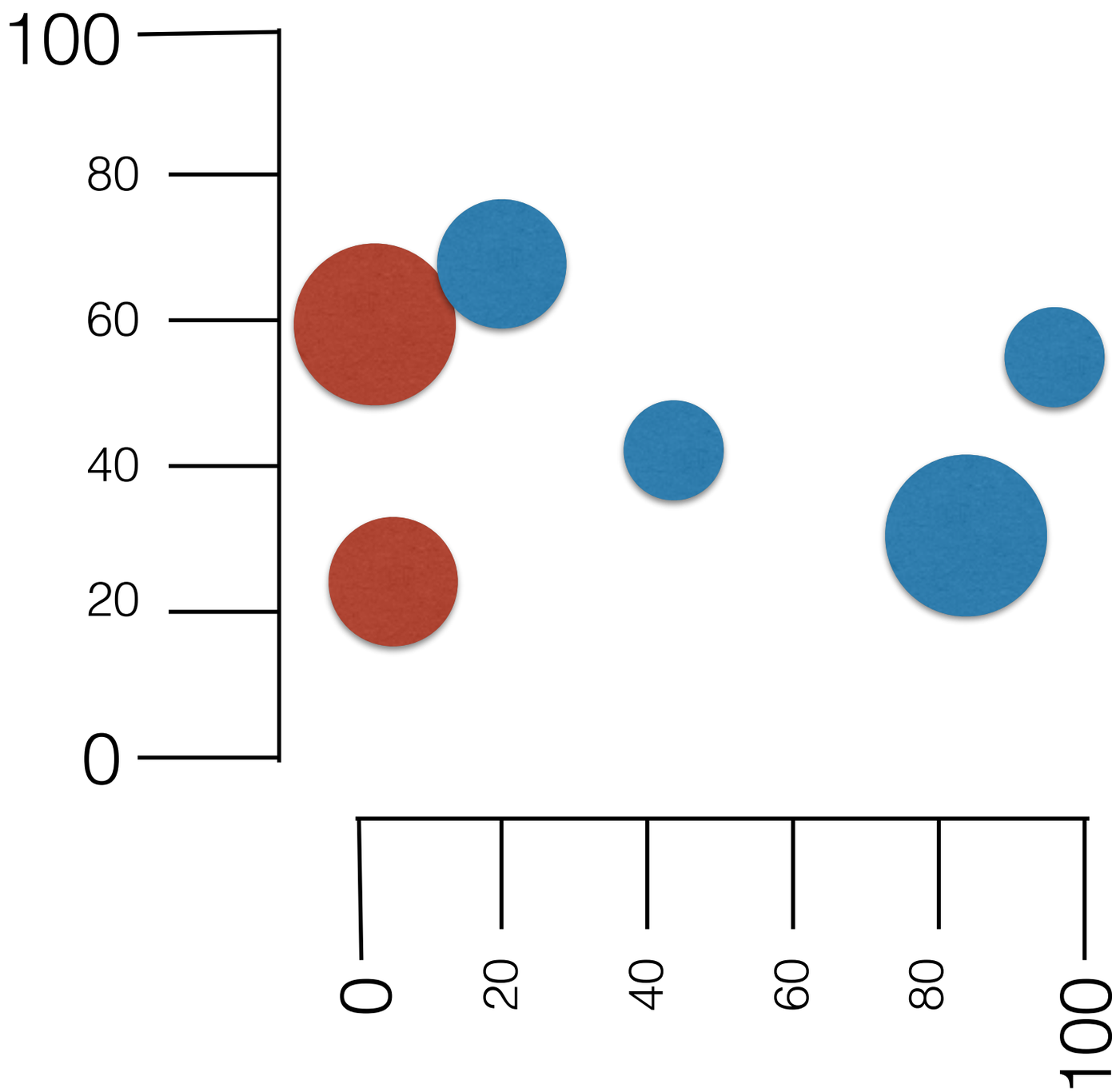
<g id=‘blob’></g>



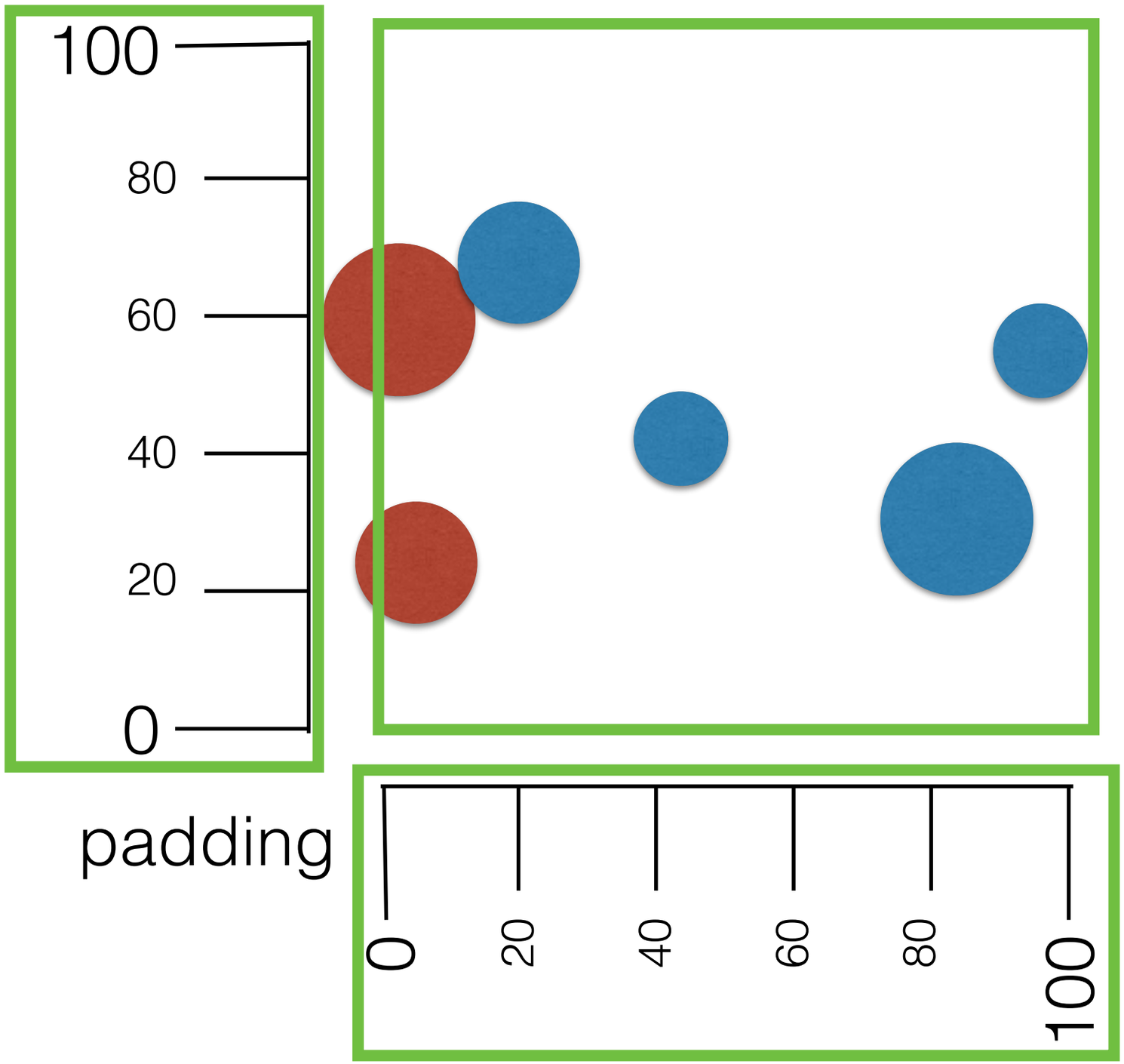

```
d3.select('#blob').call(axis);
```

```
d3.select(‘#blob’).append(‘text’)…
```

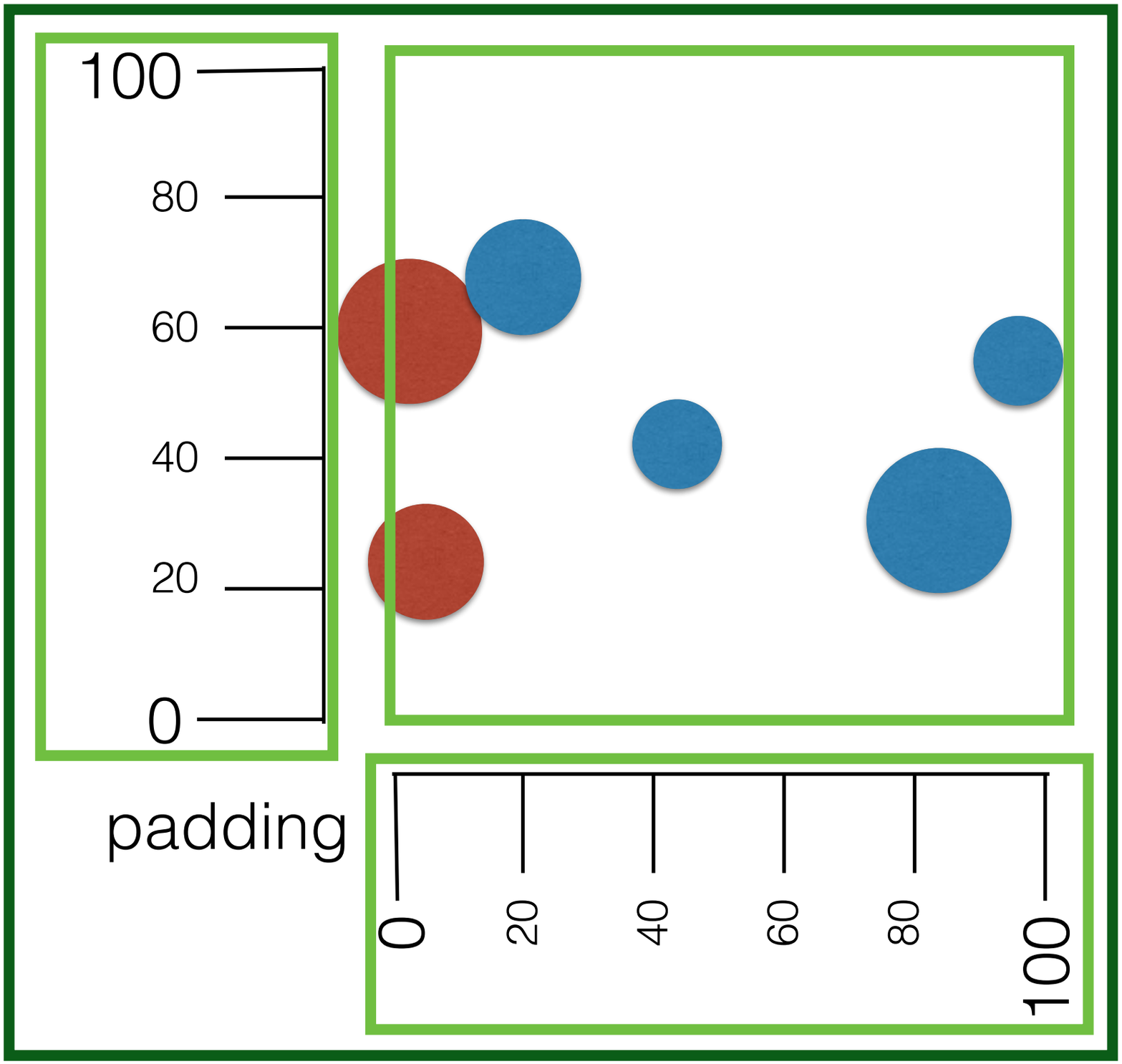
SVG Mantra: “Groups are my friend. I can structure my drawing with them.”



SVG Mantra: “Groups are my friend. I can structure my drawing with them.”



SVG Mantra: “Groups are my friend. I can structure my drawing with them.”



move the whole scatterplot

READ THE MANUAL

Follow the step by step instructions
before you start with the activities.

CS
171

Enter, Update, Exit

Lab 5

```
var x = [‘a’,’b’,’c’].forEach(function(d, i){console.log(d);}
```

```
var x = [‘a’,’b’,’c’].forEach(function(d, i){console.log(d);}

var y = [‘a’,’b’,’c’].map(function(d, i){return d+’_’+i;}
```

```
var x = [‘a’,’b’,’c’].forEach(function(d, i){console.log(d);}

var y = [‘a’,’b’,’c’].map(function(d, i){return d+’_’+i;}
```

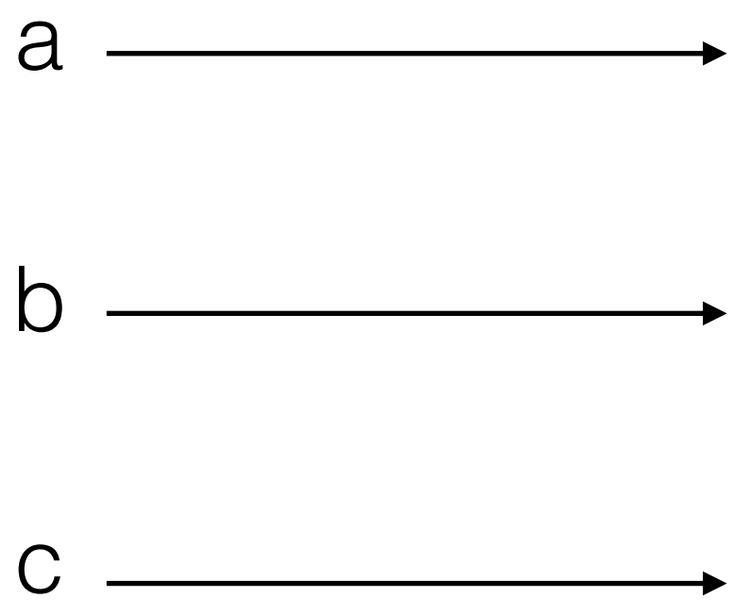
a

b

c

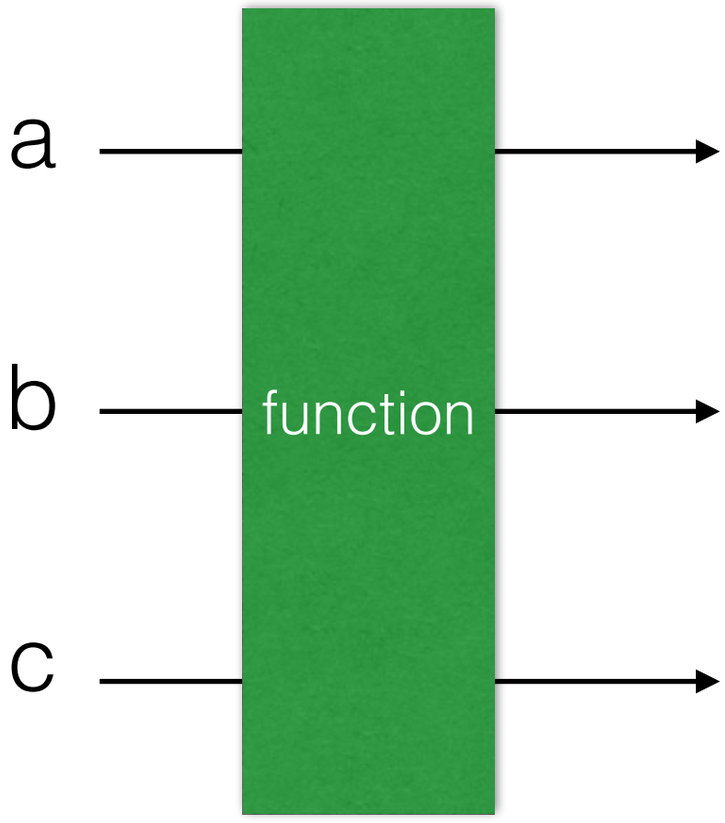
```
var x = [‘a’,’b’,’c’].forEach(function(d, i){console.log(d);}

var y = [‘a’,’b’,’c’].map(function(d, i){return d+’_’+i;}
```



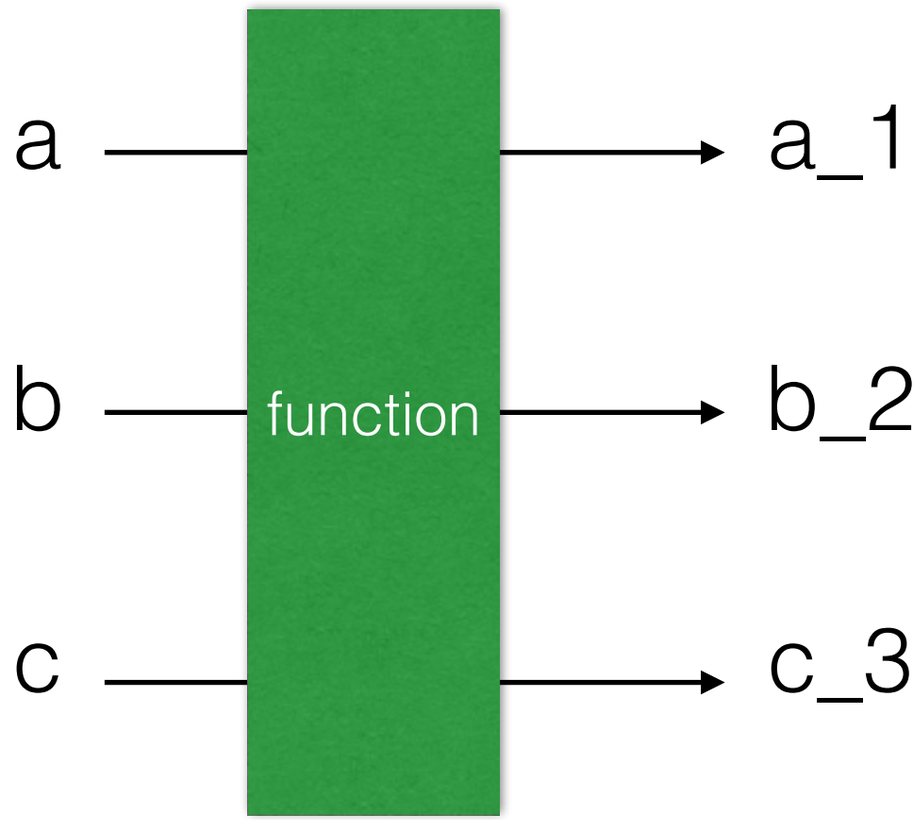
```
var x = [‘a’,’b’,’c’].forEach(function(d, i){console.log(d);}

var y = [‘a’,’b’,’c’].map(function(d, i){return d+’_’+i;}
```



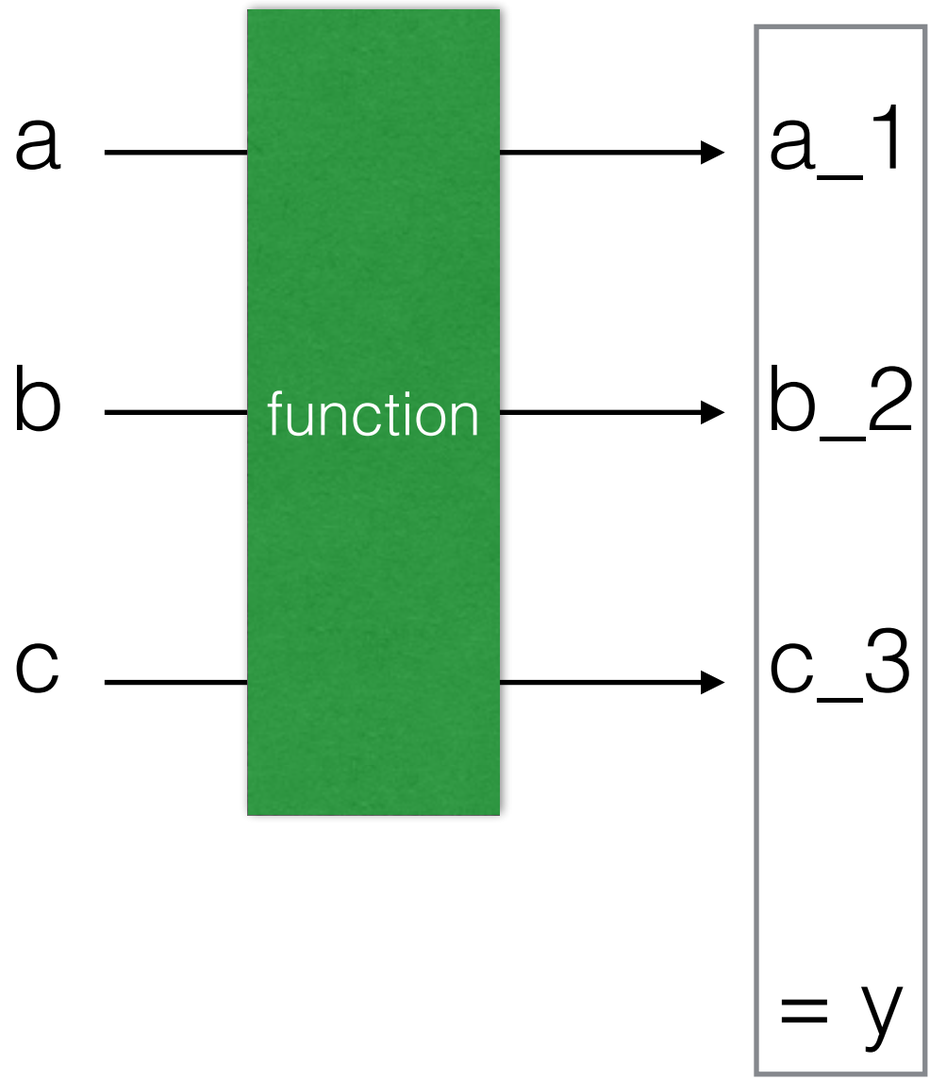
```
var x = [‘a’,’b’,’c’].forEach(function(d, i){console.log(d);}

var y = [‘a’,’b’,’c’].map(function(d, i){return d+’_’+i;}
```



```
var x = [‘a’,’b’,’c’].forEach(function(d, i){console.log(d);}

var y = [‘a’,’b’,’c’].map(function(d, i){return d+’_’+i;}
```



Data Driven Documents (D3):
Every data item maps to a visual item.

homer

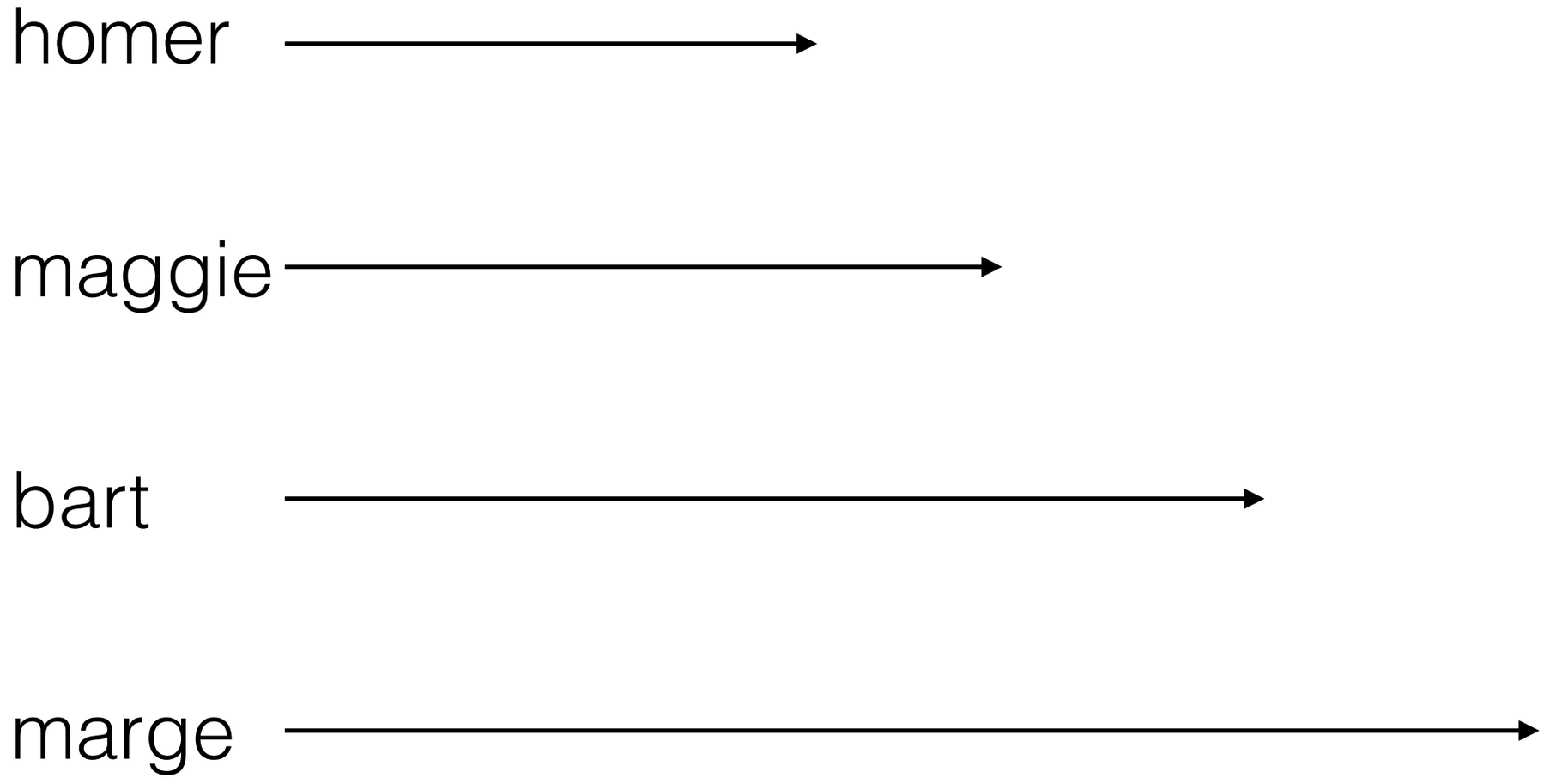
maggie

bart

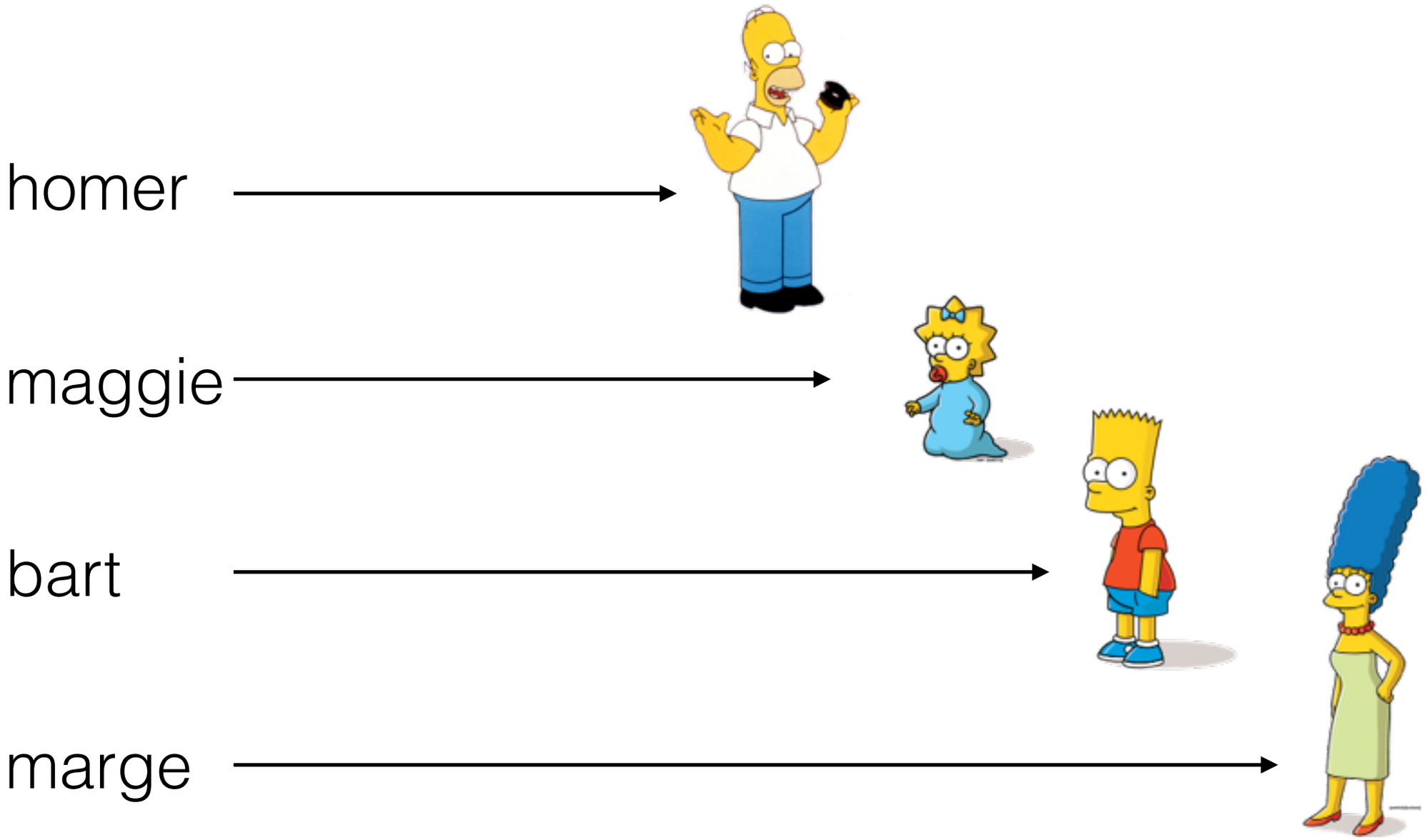
marge

Data **Document (Vis)**

Data Driven Documents (D3):
Every data item maps to a visual item.

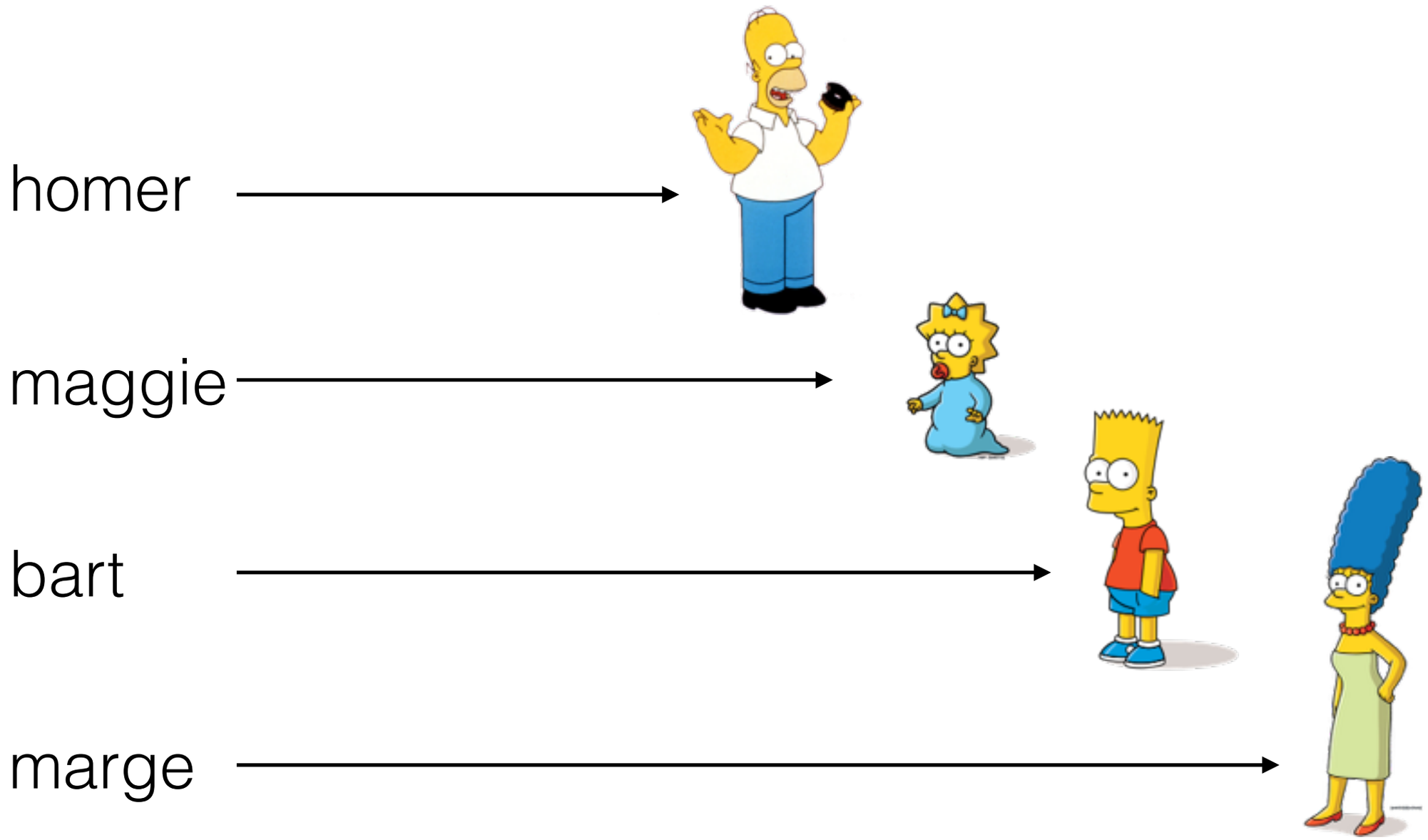


Data Driven Documents (D3):
Every data item maps to a visual item.



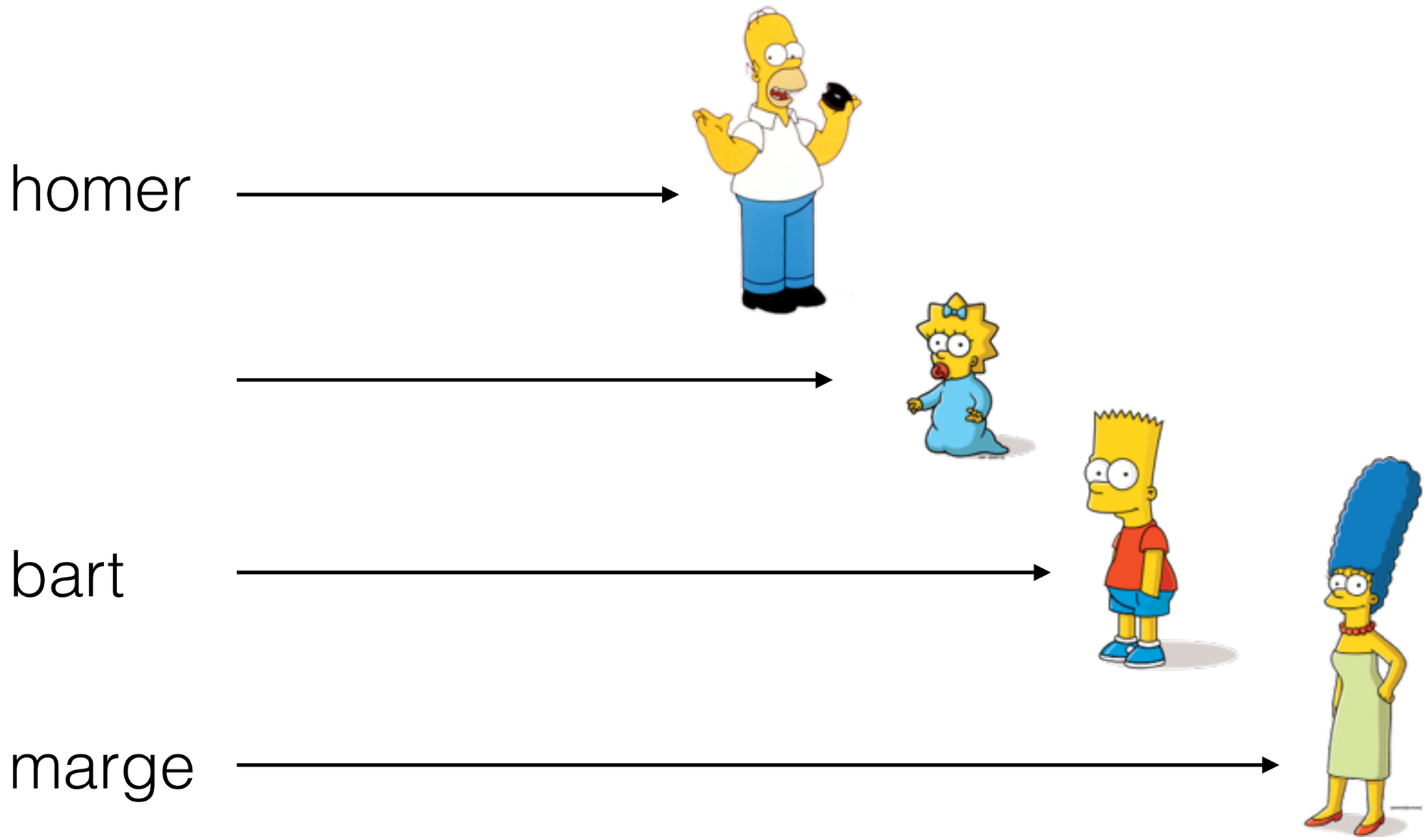
Data **Document (Vis)**

Data Driven Documents (D3):
If data disappears…



Data **Document (Vis)**

Data Driven Documents (D3): If data disappears…



Data Driven Documents (D3): If data disappears…

homer

bart

marge

Data **Document (Vis)**

Data Driven Documents (D3):
If data appears…

homer

bart

marge

Data **Document (Vis)**

Data Driven Documents (D3):
If data appears…

homer

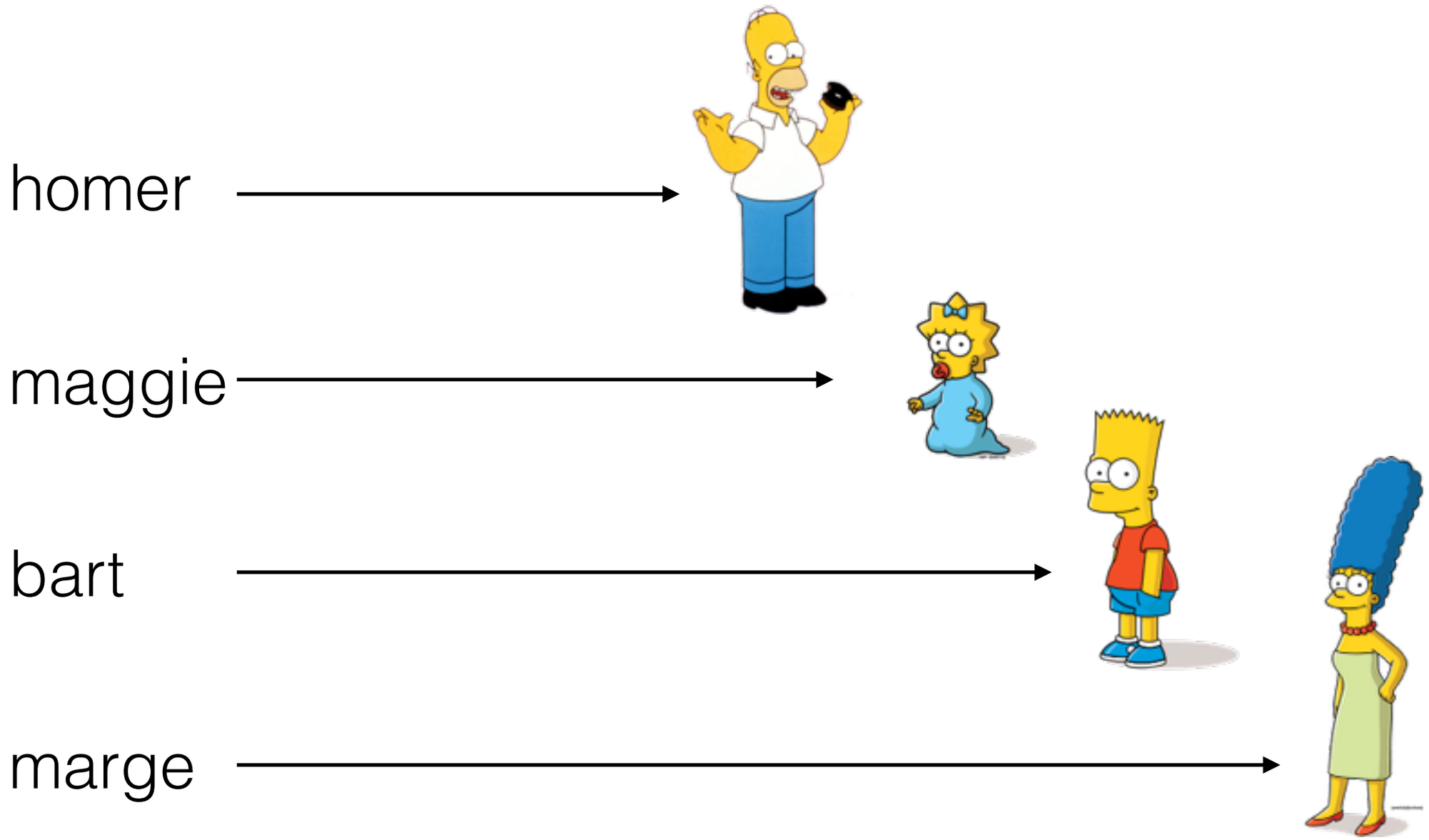
maggie

bart

marge

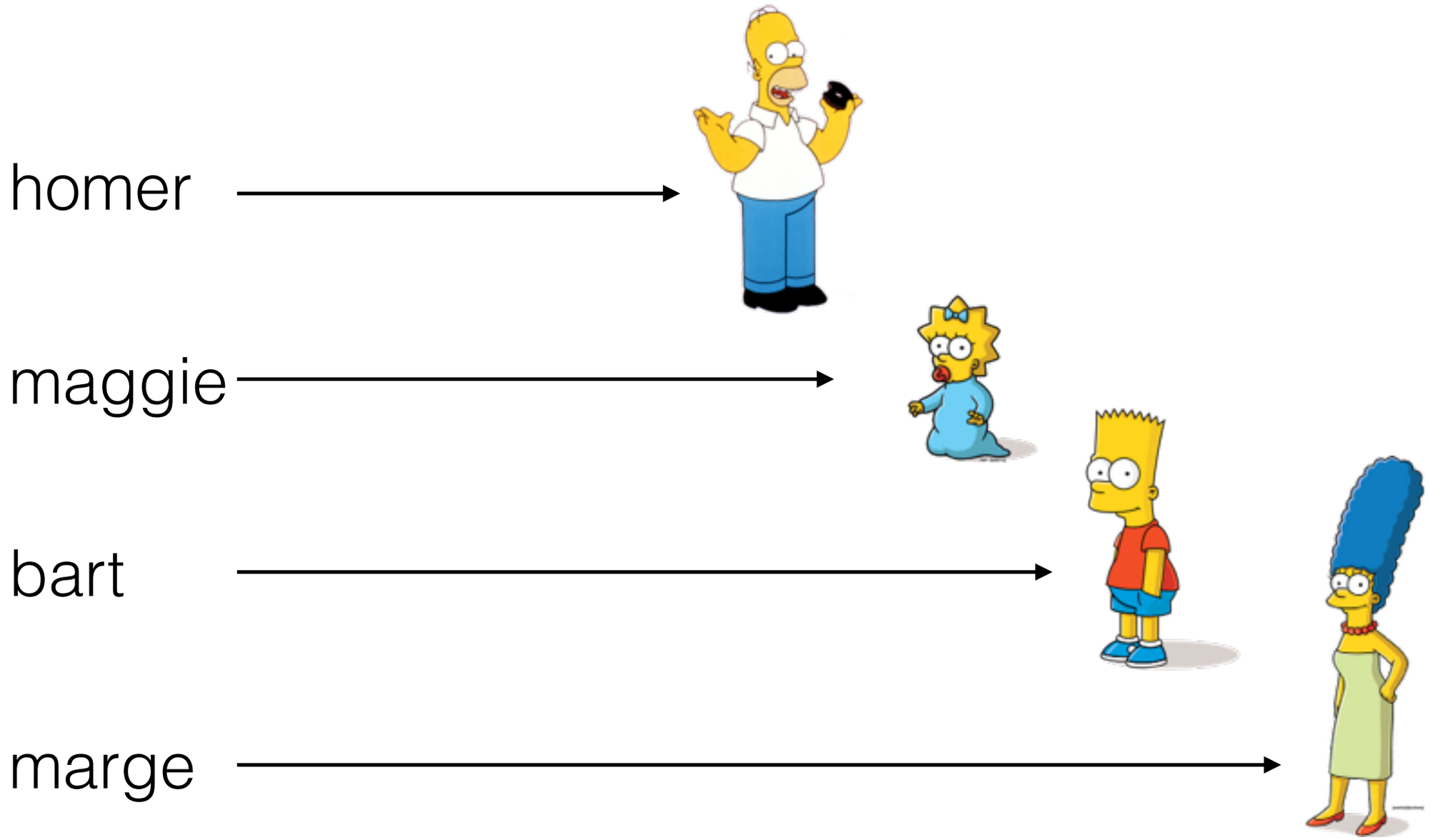
Data **Document (Vis)**

Data Driven Documents (D3):
If data appears…



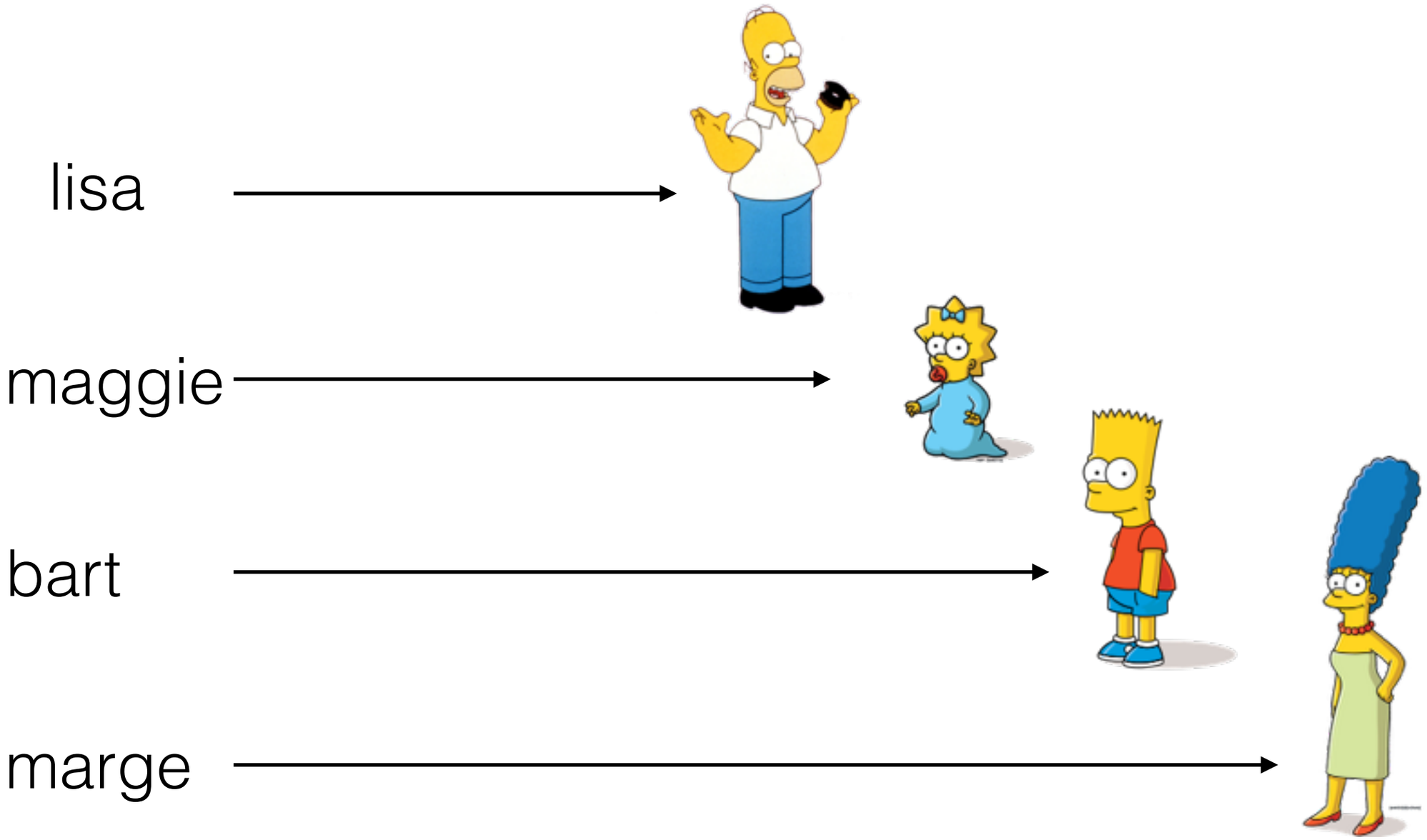
Data **Document (Vis)**

Data Driven Documents (D3):
If data changes…

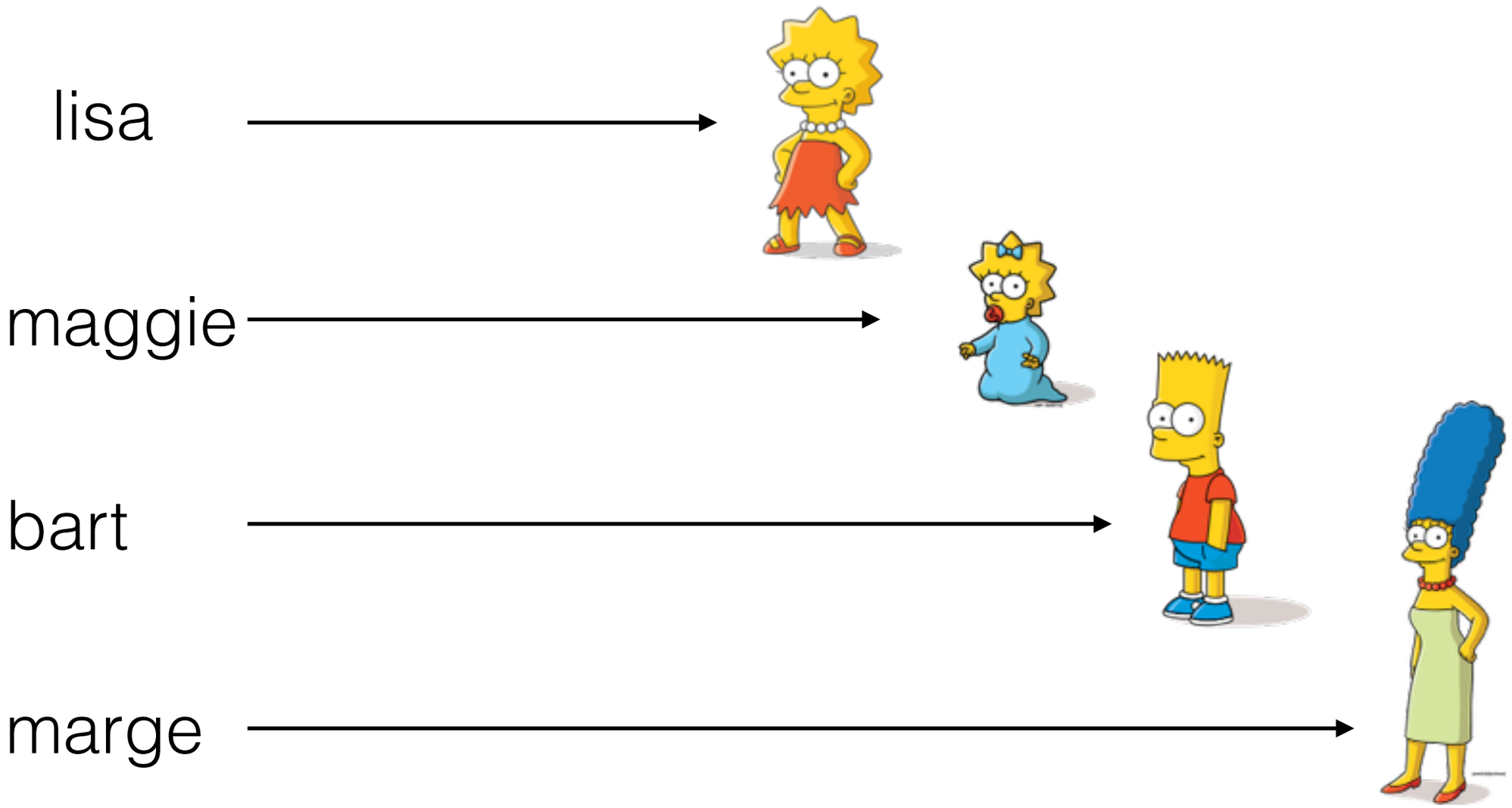


Data **Document (Vis)**

Data Driven Documents (D3):
If data changes…

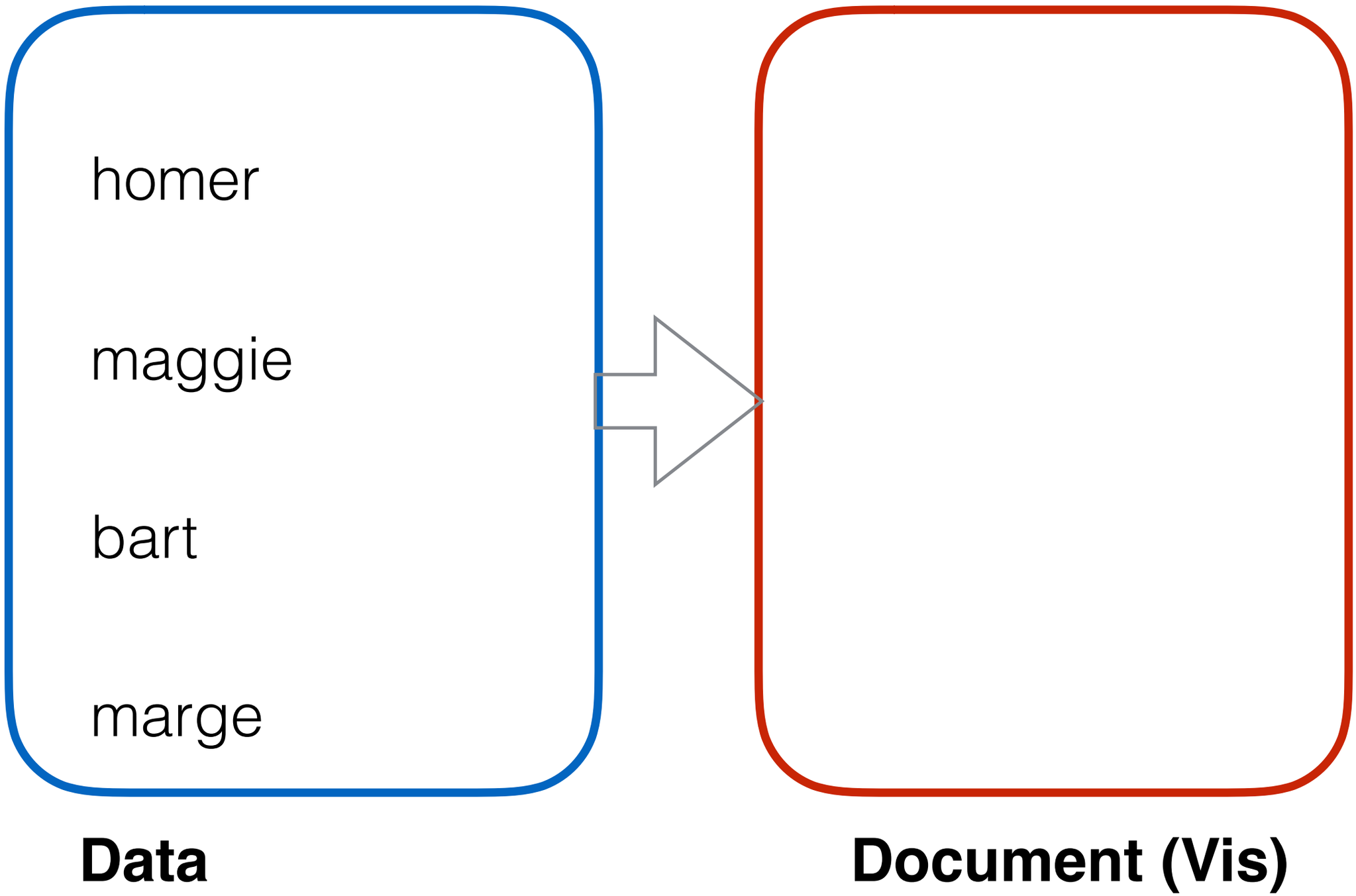


Data Driven Documents (D3):
If data changes…

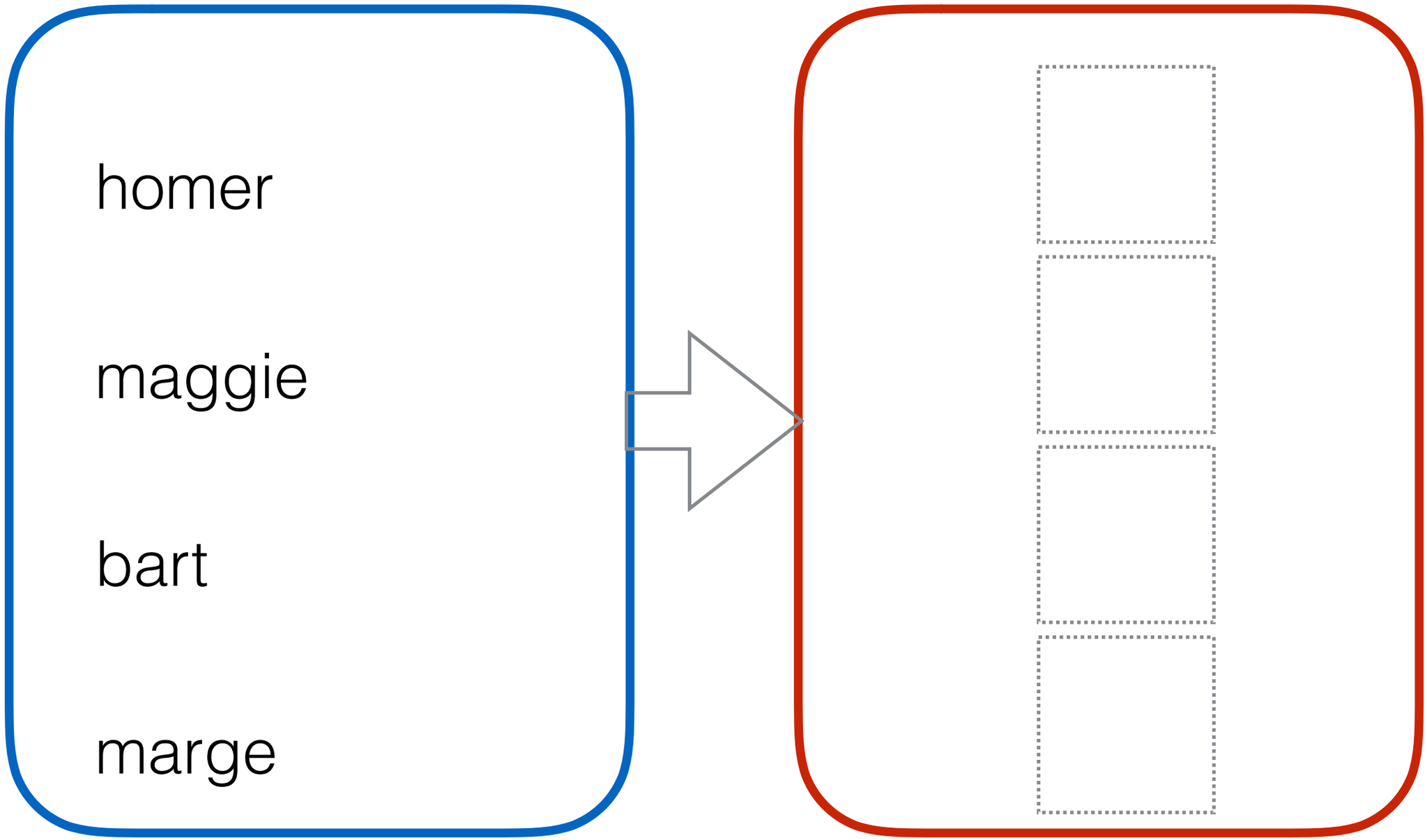


Data **Document (Vis)**

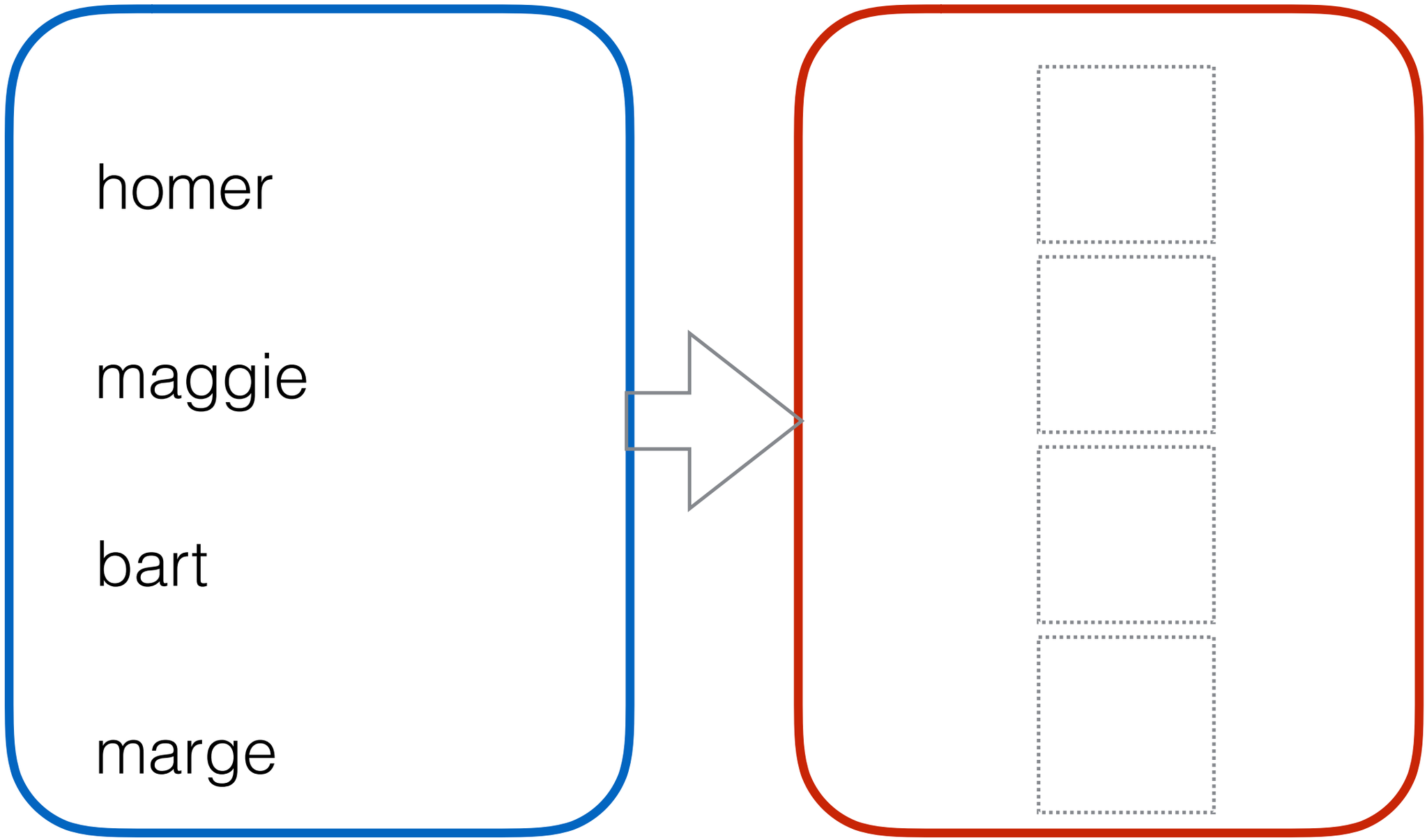
```
var allData = {‘homer’,’maggie’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
```



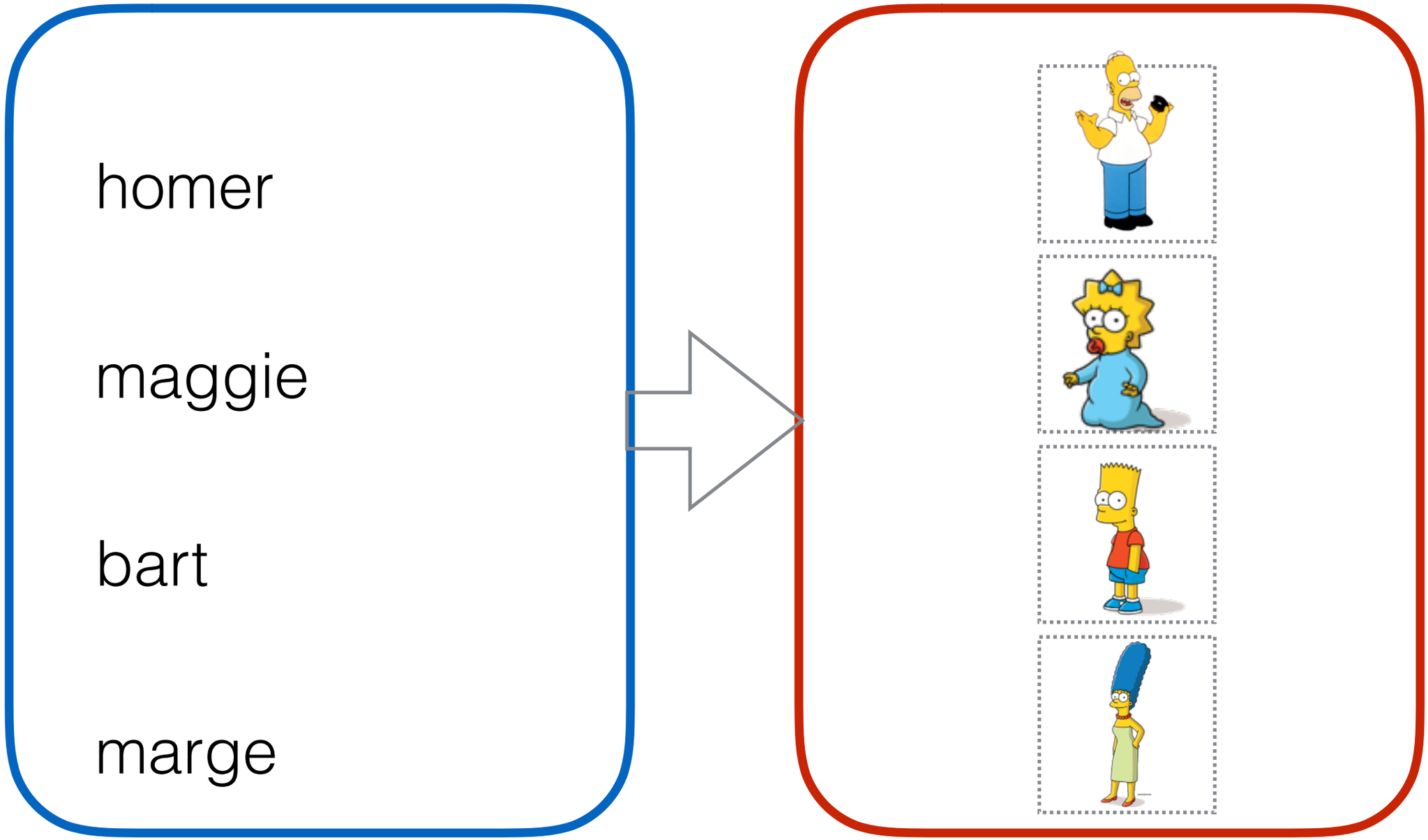
```
var allData = {‘homer’,’maggie’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
var allThatEnter = simpsons.enter()
```



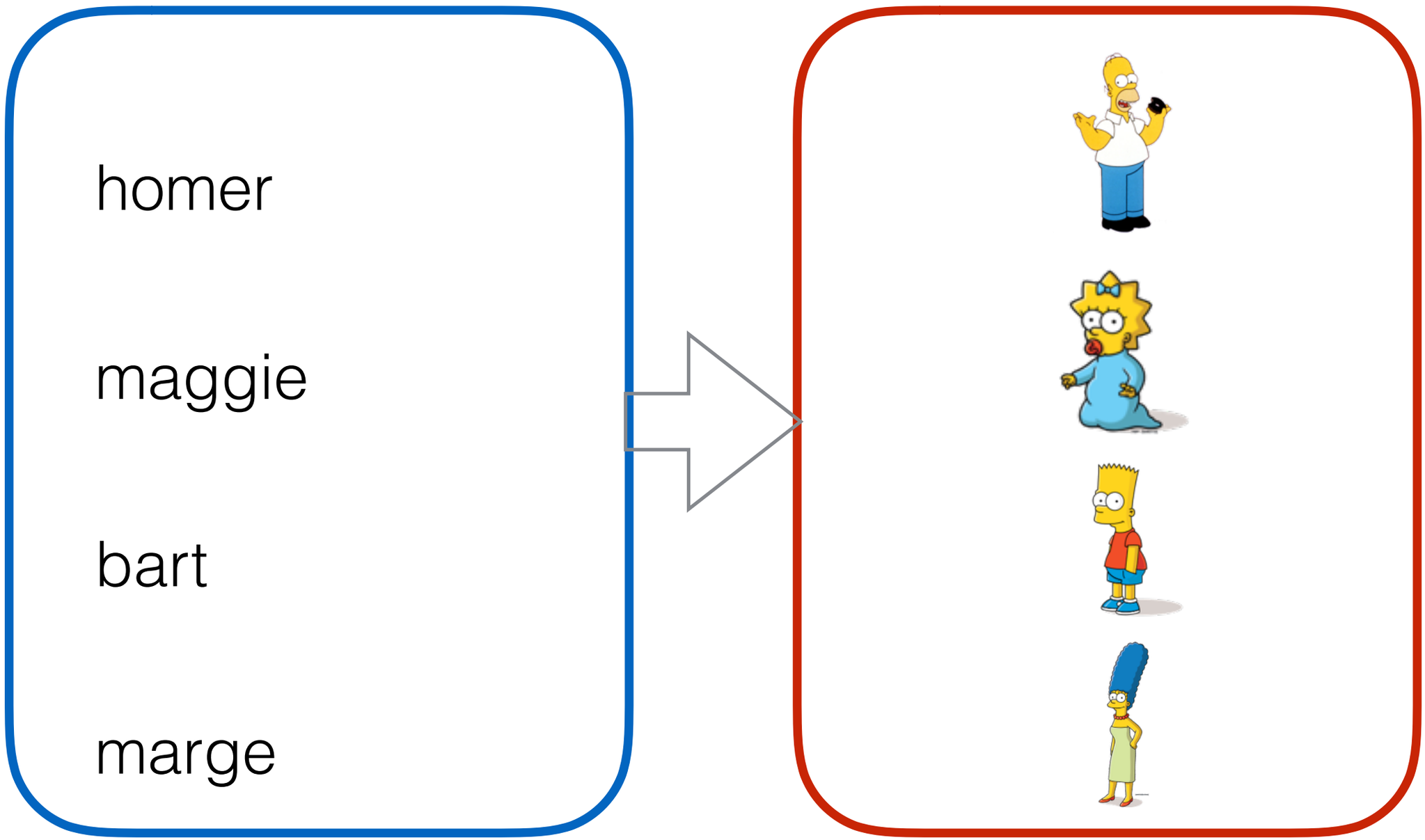
```
var simpsons = svg.selectAll(“.simpson”).data(allData);
var allThatEnter = simpsons.enter()
allThatEnter.append(‘img’).attr(…);
```



```
var simpsons = svg.selectAll(“.simpson”).data(allData);
var allThatEnter = simpsons.enter()
allThatEnter.append(‘img’).attr(…);
```

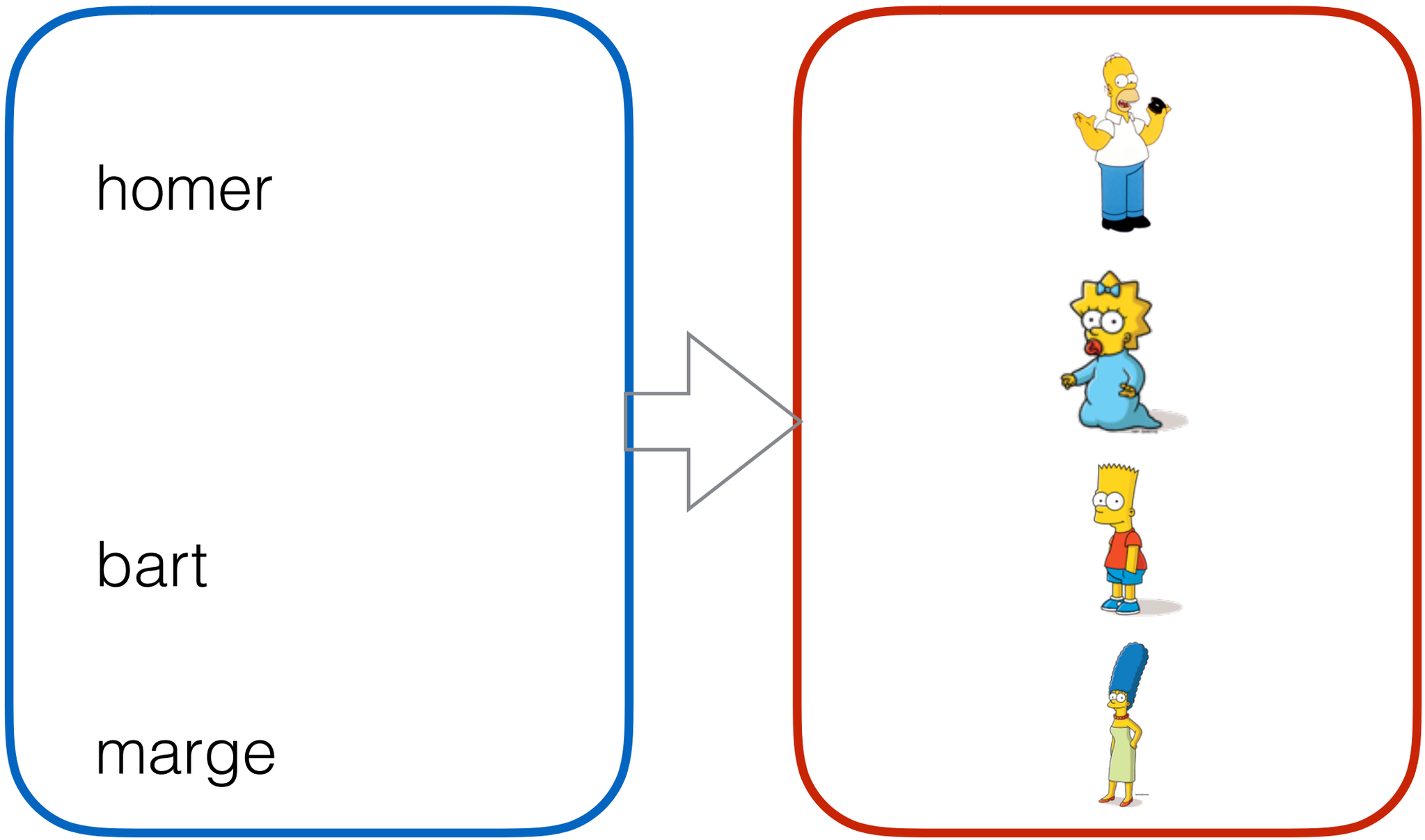


```
var allData = {‘homer’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
```



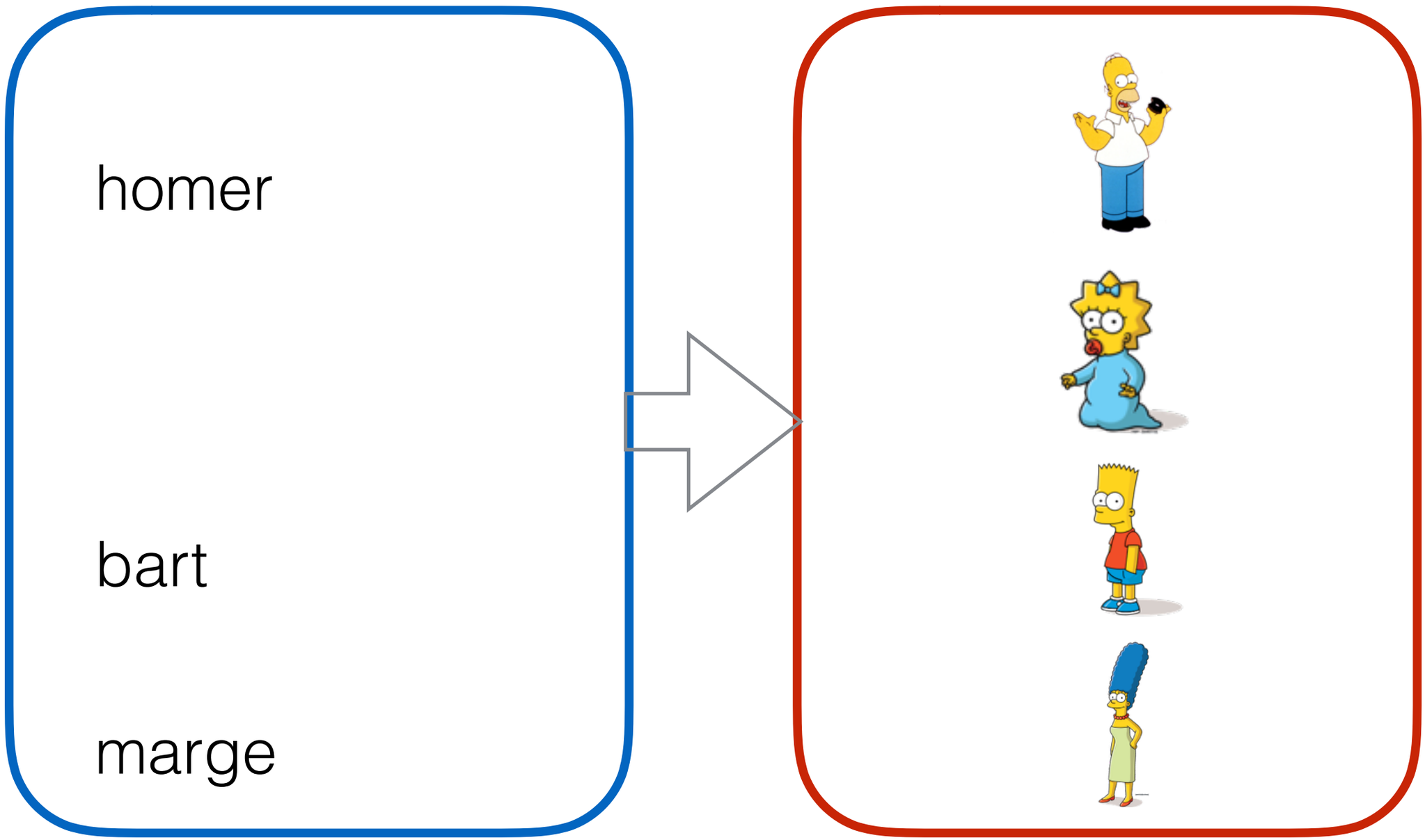
Data **Document (Vis)**

```
var allData = {‘homer’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
```



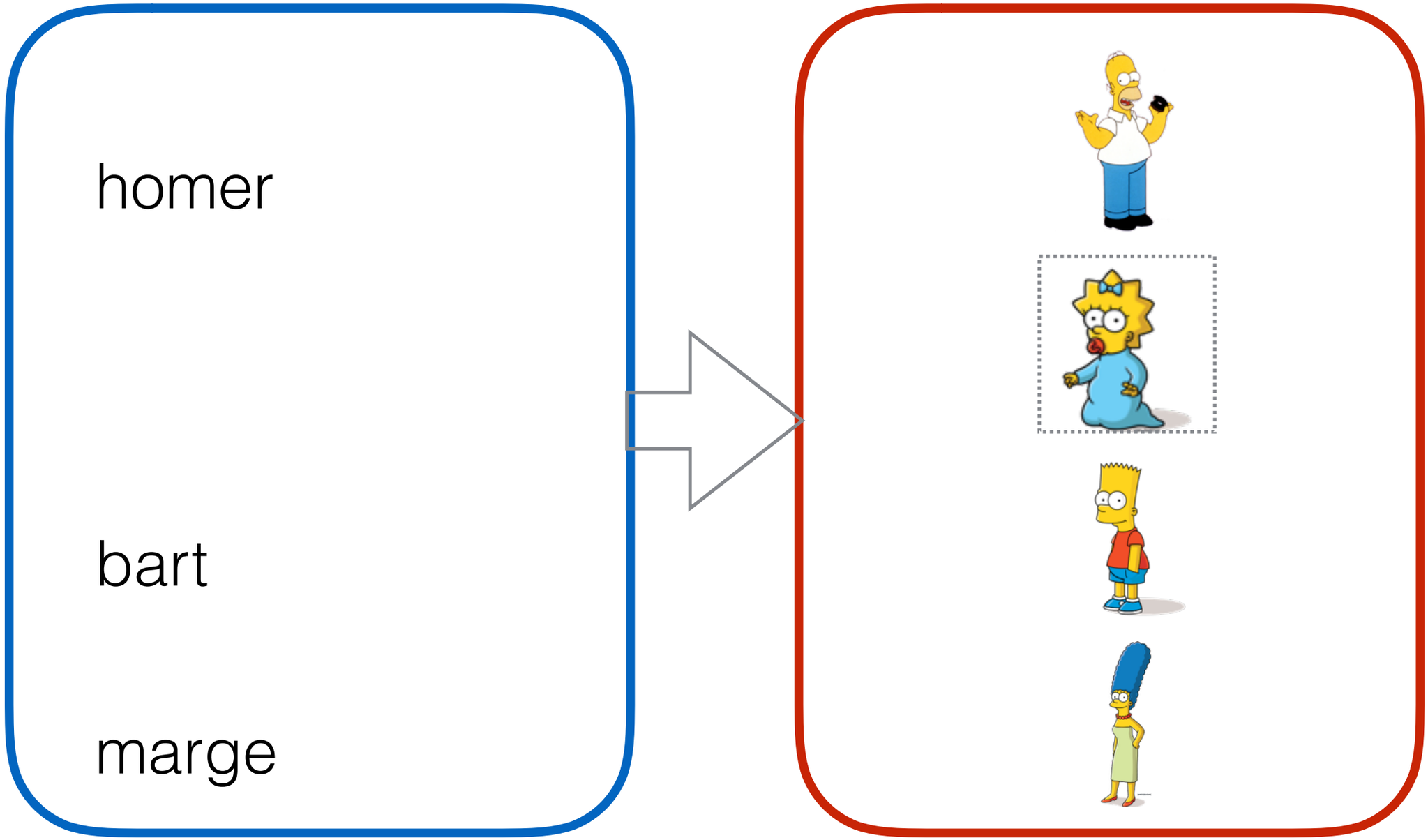
Data **Document (Vis)**

```
var allData = {‘homer’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
var allThatLeave = simpsons.exit()
```

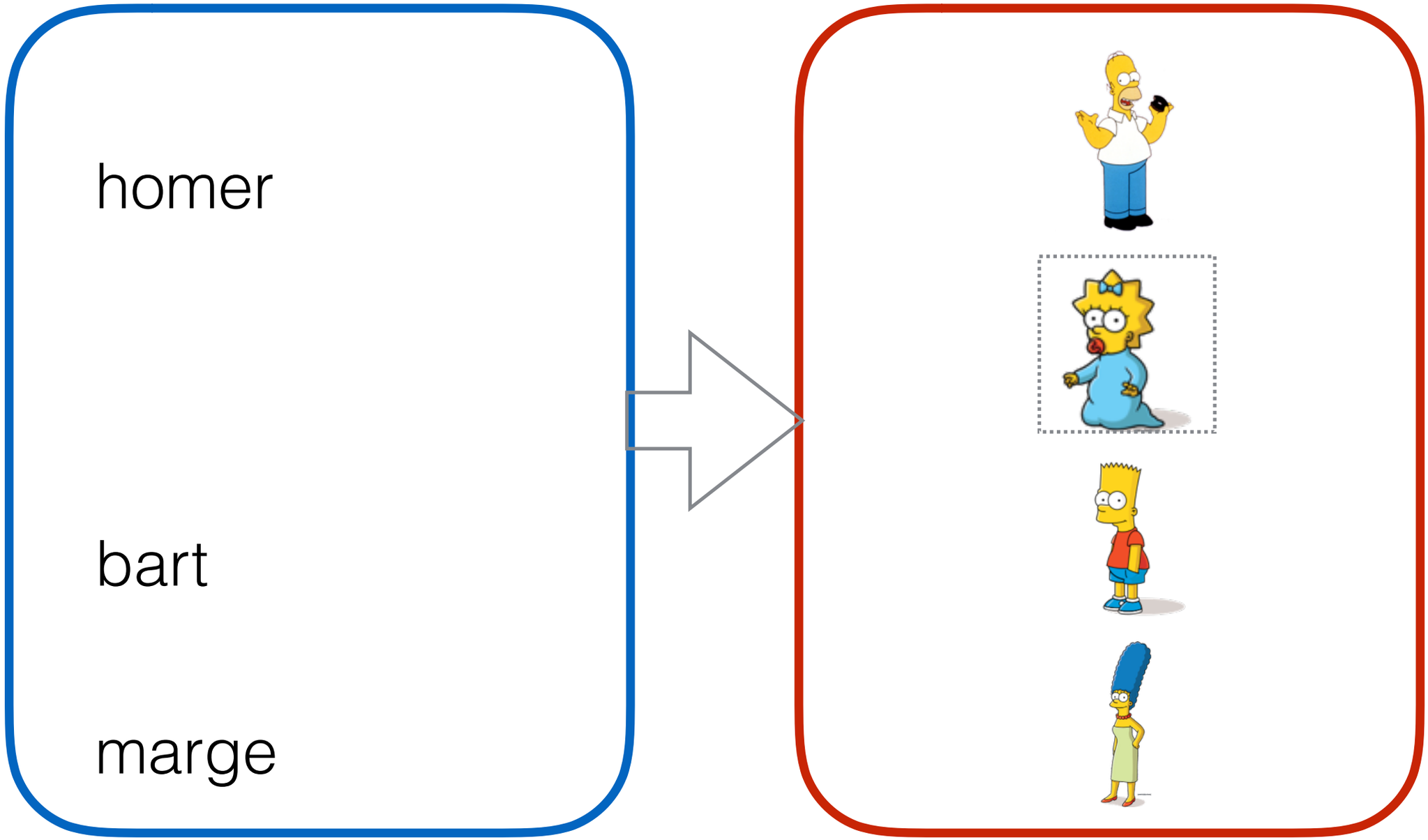


Data **Document (Vis)**

```
var allData = {‘homer’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
var allThatLeave = simpsons.exit()
```

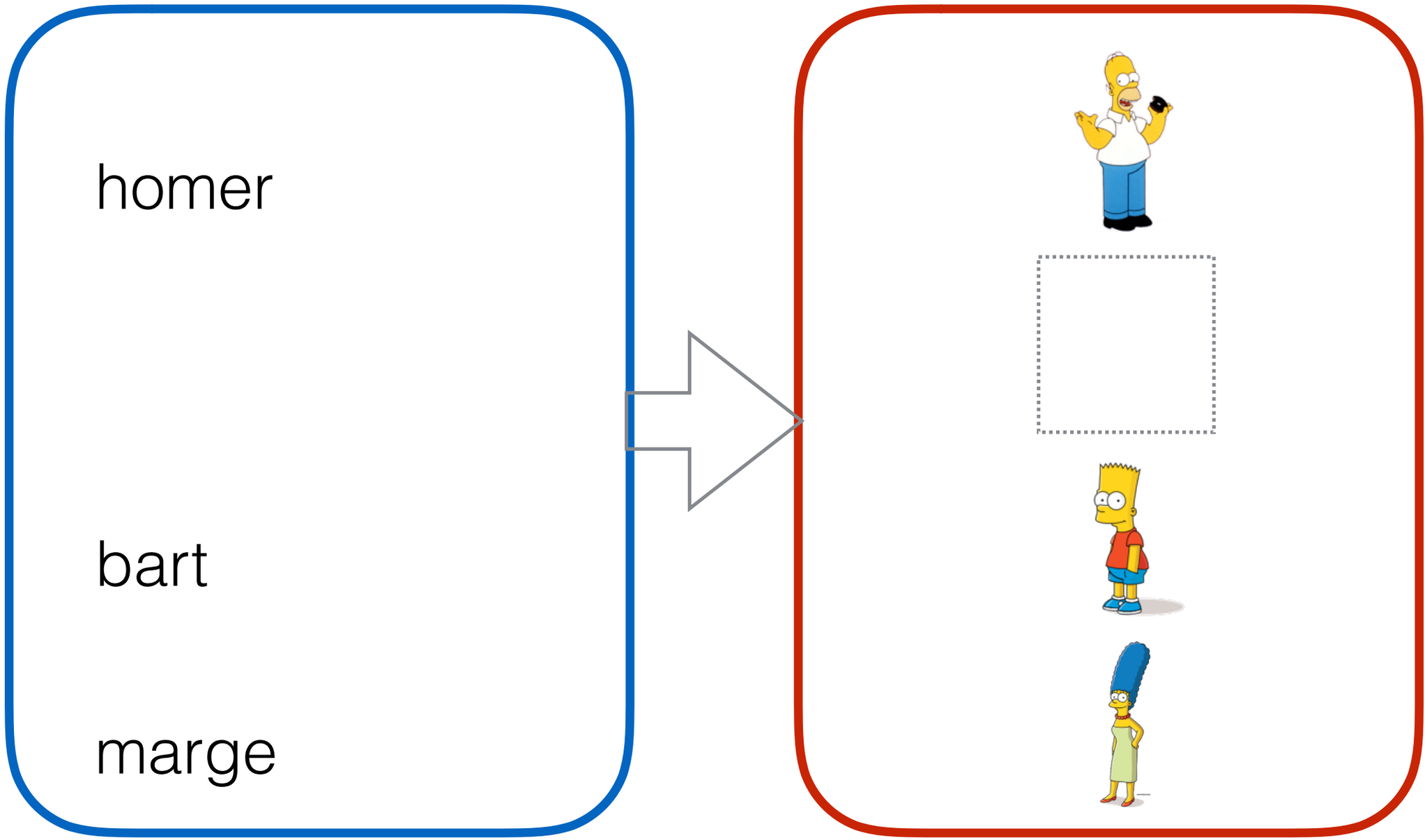


```
var allData = {‘homer’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
var allThatLeave = simpsons.exit();
allThatLeave.remove();
```

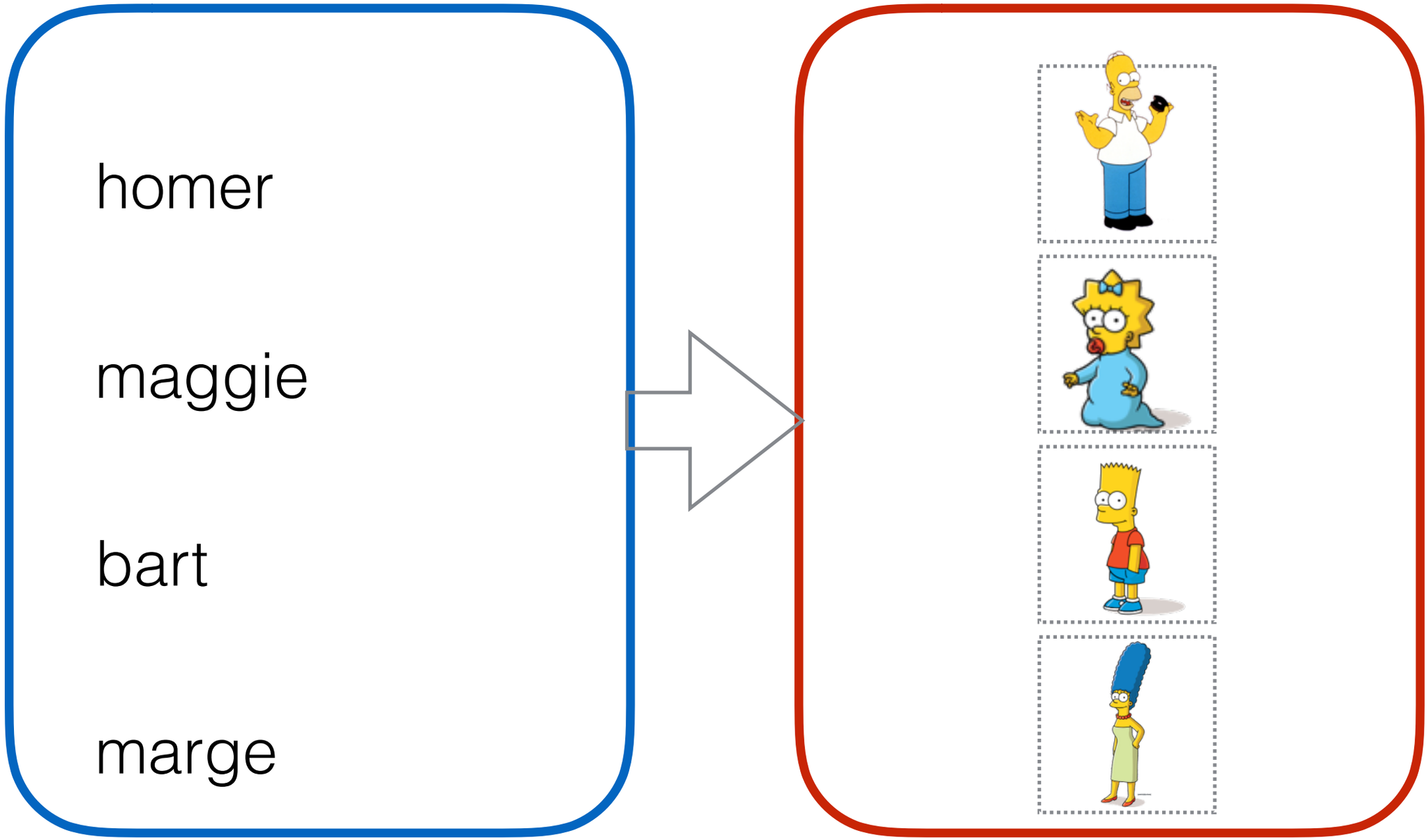


Data **Document (Vis)**

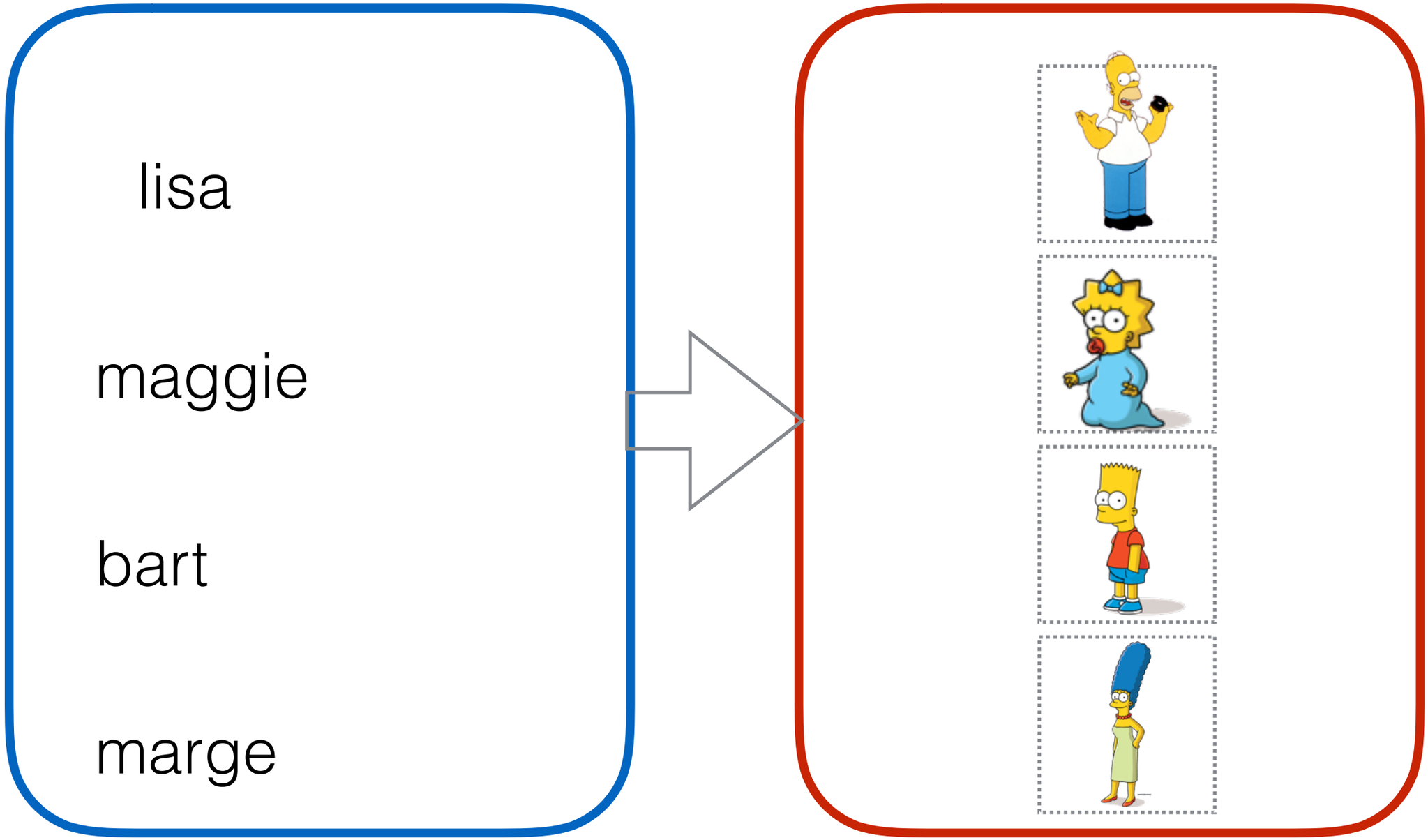
```
var allData = {‘homer’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
var allThatLeave = simpsons.exit();
allThatLeave.remove();
```



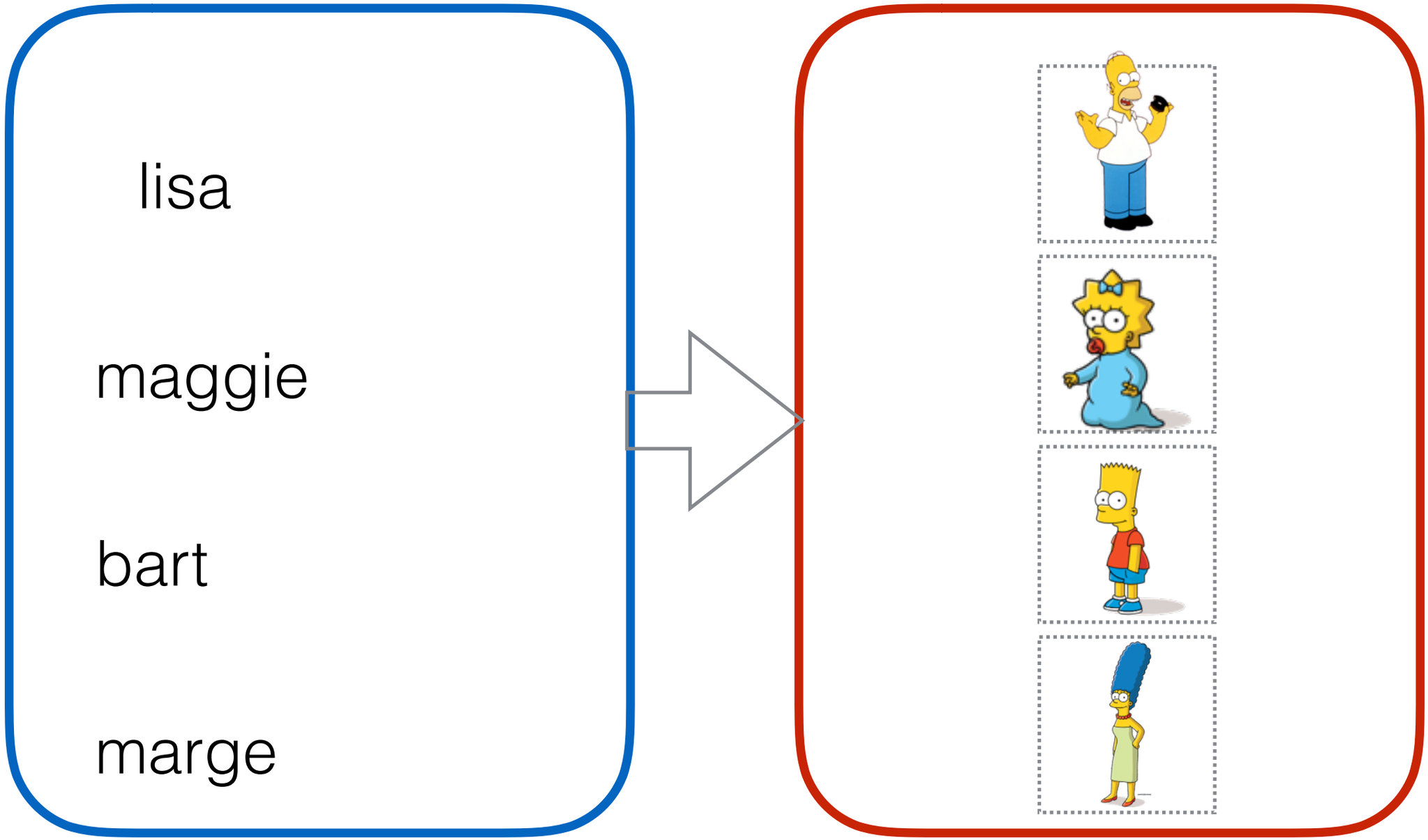
```
var allData = {‘lisa’,’maggie’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
```



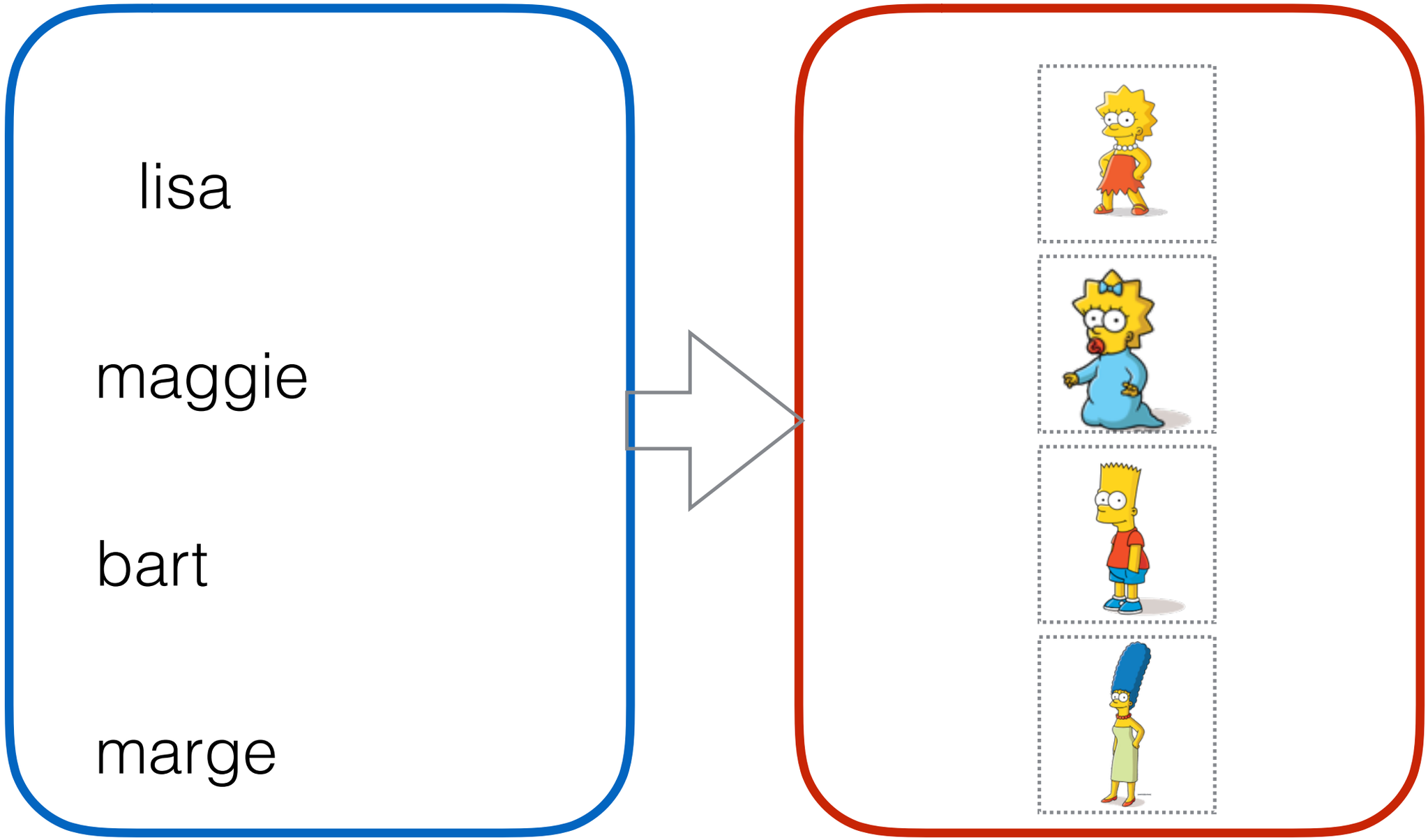
```
var allData = {‘lisa’,’maggie’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
```



```
var allData = {‘lisa’,’maggie’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
simpsons.attr(‘href’,function(d){return d+”.jpg”};
```



```
var allData = {‘lisa’,’maggie’,’bart’,’marge’}
var simpsons = svg.selectAll(“.simpson”).data(allData);
simpsons.attr(‘href’,function(d){return d+”.jpg”};
```



Hen's most frequently used D3 snippet

```
var simpsons = svg.selectAll(".simpsons").data(allData);
simpsons.exit().remove();

// --- adding Element to class simpsons
var simpsonsEnter = simpsons.enter().append("circle").attr({
    "class":"simpsons"
    "href":function(d){return d+".jpg"}
})

// --- changing nodes for simpsons
simpsons.attr({
    r: function(d,i ){return i*20;},
    cx: function (d){return scalePosX(d);}
})
```

Hen's most frequently used D3 snippet

```
var simpsons = svg.selectAll(".simpsons").data(allData);
simpsons.exit().remove();

// --- adding Element to class simpsons
var simpsonsEnter = simpsons.enter().append("circle").attr({
    "class":"simpsons"
    "href":function(d){return d+".jpg"}
})

// --- changing nodes for simpsons
simpsons.attr({
    r: function(d,i ){return i*20;},
    cx: function (d){return scalePosX(d);}
})
```

binding

Hen's most frequently used D3 snippet

```
var simpsons = svg.selectAll(".simpsons").data(allData);
simpsons.exit().remove();

// --- adding Element to class simpsons
var simpsonsEnter = simpsons.enter().append("circle").attr({
    "class":"simpsons"
    "href":function(d){return d+".jpg"}
})

// --- changing nodes for simpsons
simpsons.attr({
    r: function(d,i ){return i*20;},
    cx: function (d){return scalePosX(d);}
})
```

binding

remove

Hen's most frequently used D3 snippet

```
var simpsons = svg.selectAll(".simpsons").data(allData);
```

binding

```
simpsons.exit().remove();
```

remove

```
// --- adding Element to class simpsons
var simpsonsEnter = simpsons.enter().append("circle").attr({
    "class":"simpsons"
    "href":function(d){return d+".jpg"}
})
```

add

define all **static** attributes

```
// --- changing nodes for simpsons
simpsons.attr({
    r: function(d,i ){return i*20;},
    cx: function (d){return scalePosX(d);}
})
```

Hen's most frequently used D3 snippet

```
var simpsons = svg.selectAll(".simpsons").data(allData);
simpsons.exit().remove();

// --- adding Element to class simpsons
var simpsonsEnter = simpsons.enter().append("circle").attr({
    "class":"simpsons"
    "href":function(d){return d+".jpg"}
})

// --- changing nodes for simpsons
simpsons.attr({
    r: function(d,i ){return i*20;},
    cx: function (d){return scalePosX(d);}
})
```

binding

remove

add

define all **static** attributes

every data item has a visual representative **AND**
every visual representatives has a data item

Hen's most frequently used D3 snippet

```
var simpsons = svg.selectAll(".simpsons").data(allData);
```
binding

```
simpsons.exit().remove();
```
remove

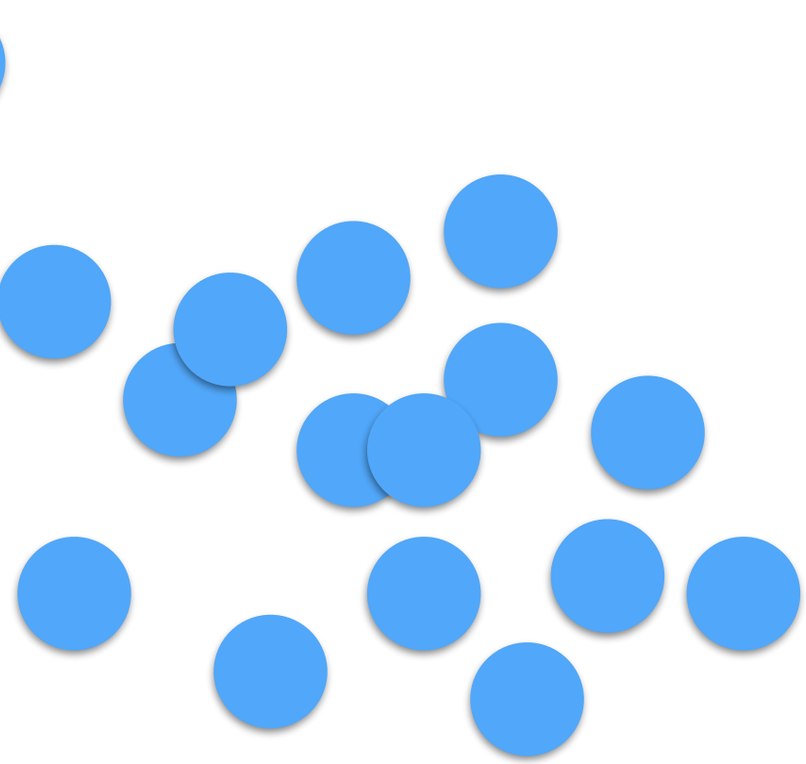
```
// --- adding Element to class simpsons
var simpsonsEnter = simpsons.enter().append("circle").attr({
    "class":"simpsons"
    "href":function(d){return d+".jpg"}
})
```
add

define all **static** attributes

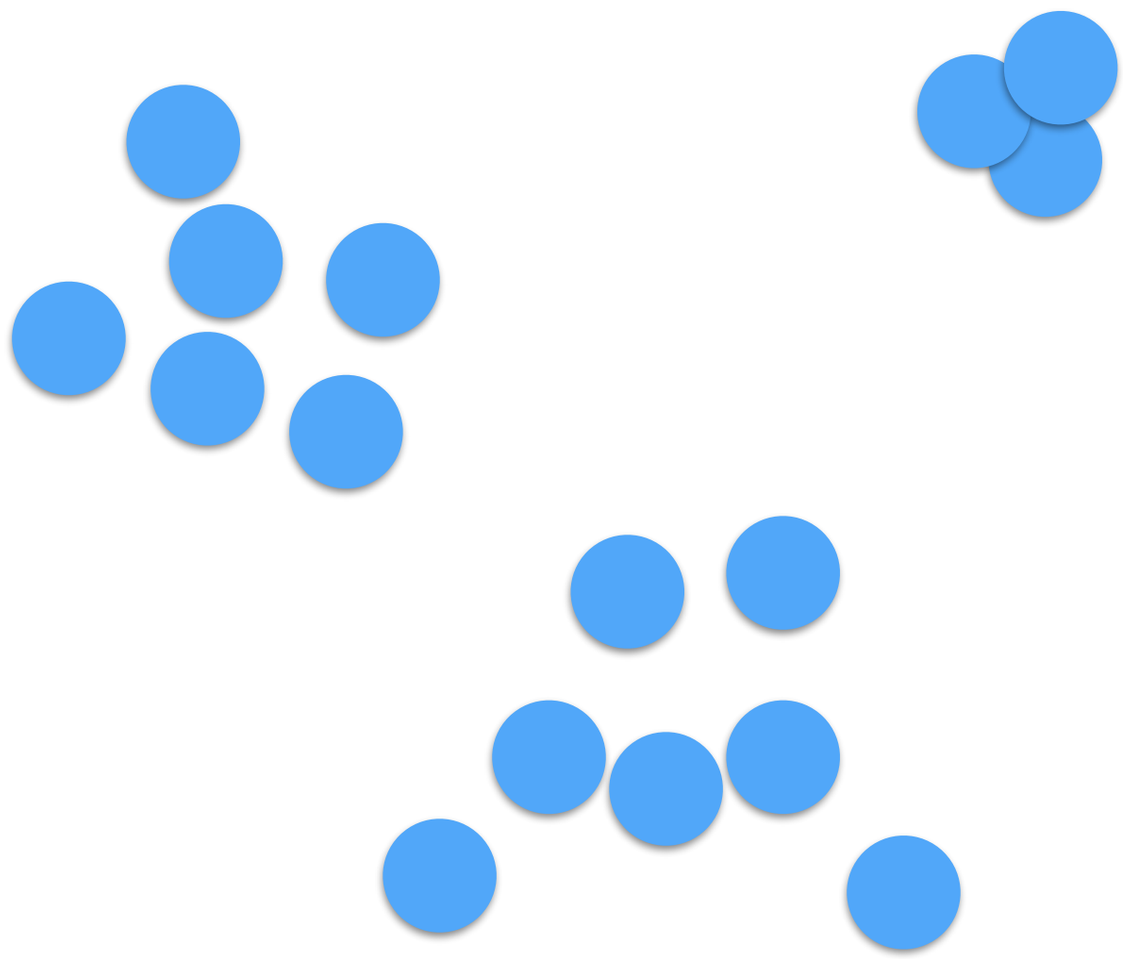
```
// --- changing nodes for simpsons
simpsons.attr({
    r: function(d,i ){return i*20;},
    cx: function (d){return scalePosX(d);}
})
```
define all **dynamic** attributes

every data item has a visual representative **AND**
every visual representatives has a data item

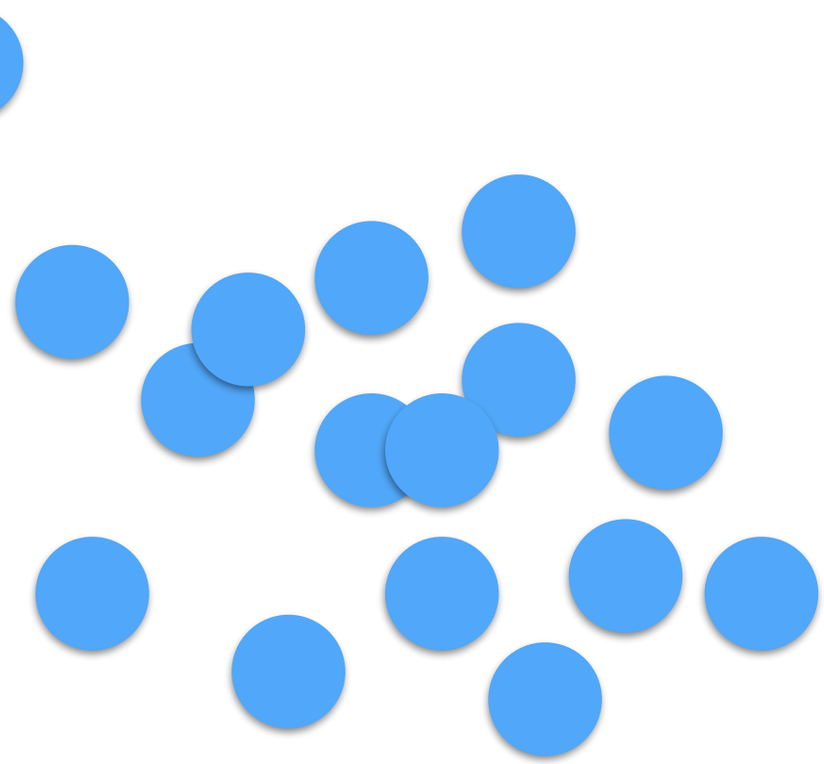
Remember Lecture 4



Remember Lecture 4

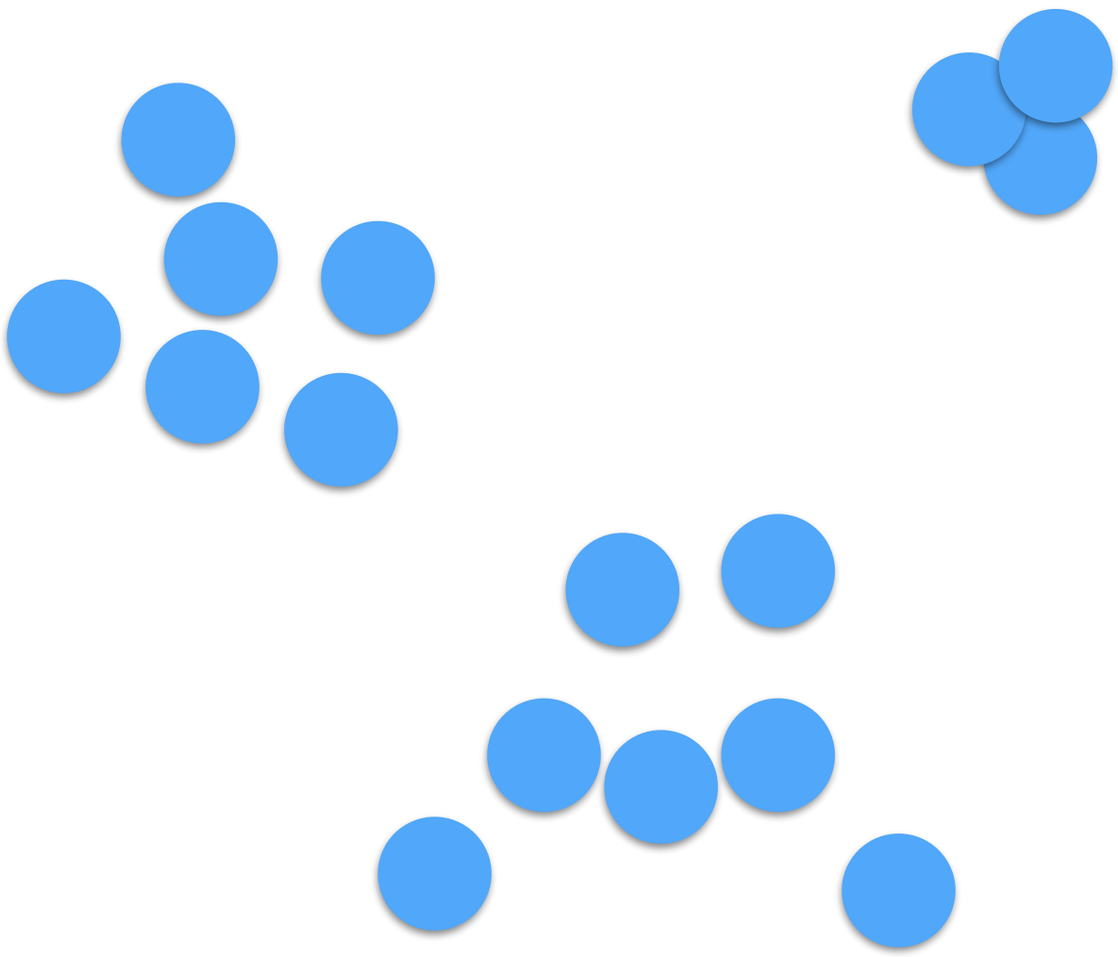


Change Blindness



transitions can help

Change Blindness



transitions can help

```
var simpsons = svg.selectAll(".simpsons").data(allData);
simpsons.exit().remove();

// ---- adding Element to class simpsons
var simpsonsEnter = simpsons.enter().append("circle").attr({
    "class":"simpsons"
    "href":function(d){return d+".jpg"}
})

// ---- changing nodes for simpsons
simpsons.attr({
    r: function(d,i ){return i*20;},
    cx: function (d){return scalePosX(d);}
})
```

.transition().duration(2000)

```
var simpsons = svg.selectAll(".simpsons").data(allData);
simpsons.exit().remove();

// ---- adding Element to class simpsons
var simpsonsEnter = simpsons.enter().append("circle").attr({
    "class":"simpsons"
    "href":function(d){return d+".jpg"}
})

// ---- changing nodes for simpsons
simpsons.transition().duration(2000).attr({
    r: function(d,i ){return i*20;},
    cx: function (d){return scalePosX(d);}
})
```

one more thing

CS
171

Use the key function in D3

```
var allData = [
  {firstName: “homer”, name:”simpson”},
  {firstName: “maggie”, name:”simpson”},
  {firstName: “bart”, name:”simpson”},
  {firstName: “marge”, name:”simpson”}
]
```

Use the key function in D3

```
var allData = [
 {firstName: “homer”, name:”simpson”},
 {firstName: “maggie”, name:”simpson”},
 {firstName: “bart”, name:”simpson”},
 {firstName: “marge”, name:”simpson”}
]
```

```
{firstName: “homer”, name:”simpson”} = {firstName: “maggie”, name:”simpson”}
```

?

Use the key function in D3

```
var allData = [
  {firstName: “homer”, name:”simpson”},
  {firstName: “maggie”, name:”simpson”},
  {firstName: “bart”, name:”simpson”},
  {firstName: “marge”, name:”simpson”}
]
```

?

{firstName: “homer”, name:”simpson”} = {firstName: “maggie”, name:”simpson”}

```
svg.selectAll(‘.simpson’)
  .data(allData, function(d,i){return d.firstName;})
```