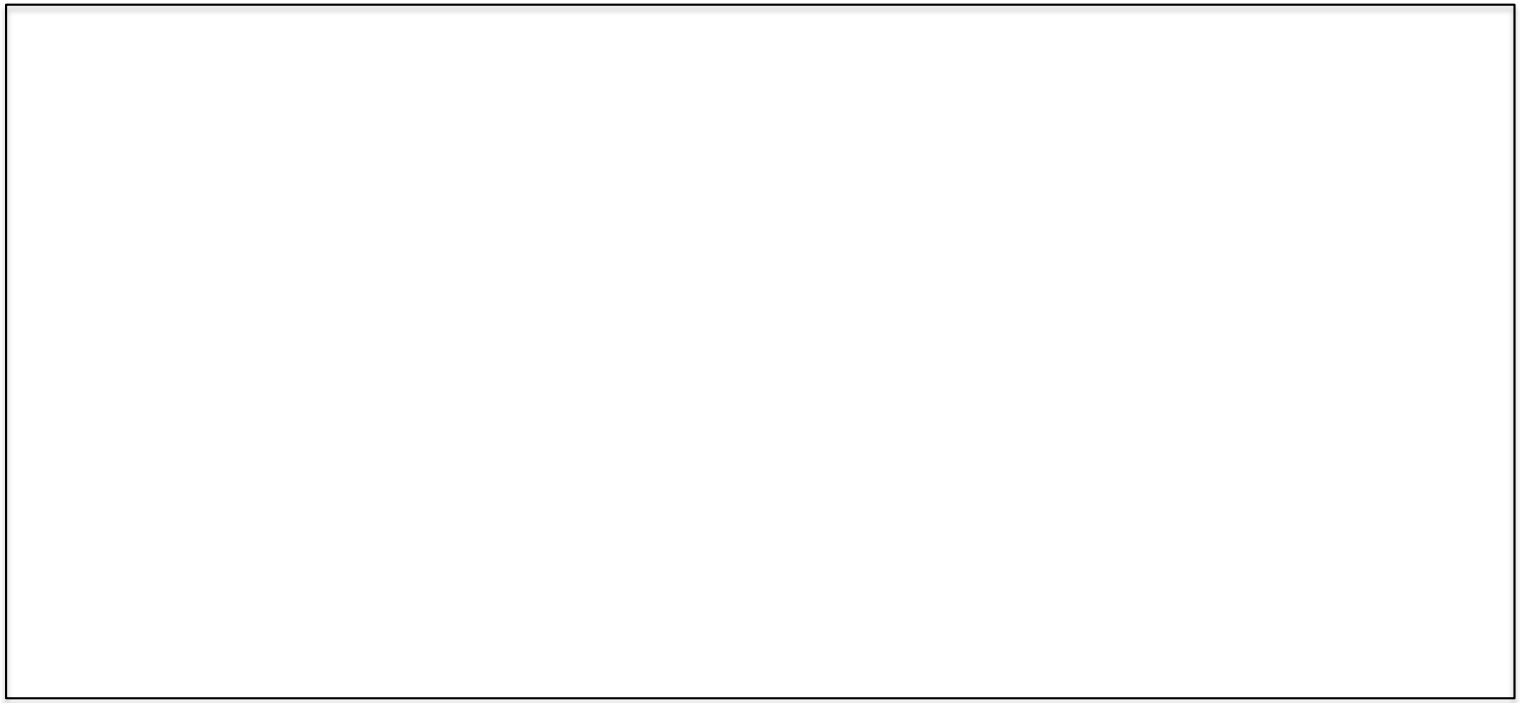
Test

1. Write a code to draw a rectangle shape on the “canvas1”



2. Select the code that draws multiple shapes (can be more than one)

- ☐ (a)
- ☐ (b)
- ☐ (c)
- ☐ (d)

(a)

```
canvas.append("circle").attr("r",20);
canvas.append("circle").attr("r",20);
```

(b)

```
array1 = [0,"text2"]
for (i = 0; i<array1.length; i++){
    canvas.append("circle").attr("r",20)
}
```

(c)

```
array1 = [5]
canvas.selectAll("circles")
        .data(array1)
        .enter()
        .append("circle").attr("r", 20)
```

(d)

```
canvas.append("circle").attr("r",20);
```

3. Move group of elements to the right by 20 points

```
var canvas1 = d3.select("body").append("svg").attr("height", 500).attr("width", 500)
    //….. Insert any code if needed bellow....



    canvas1.append("circle").attr("r",20);
    canvas1.append("circle").attr("r",20);
    //….. Insert any code if needed bellow....
```

4.

a) How many new circles will be added on the screen _____?

```
canvas.append(“circle”).attr(“r”,20);

array1 = [5, 6, 2, 1]
canvas.selectAll(“circle”)
        .data(array1)
        .enter()
        .append(“circle”).attr(“r”, 20)
```

b) How many now new circles will be added on the screen _____?

```
canvas.append(“circle”).attr(“r”,20);

array1 = [5, 6, 2, 1]
canvas.selectAll(“circlesssss”)
        .data(array1)
        .enter()
        .append(“circle”).attr(“r”, 20)
```

5.

a) How many circles will be left on the screen _____?

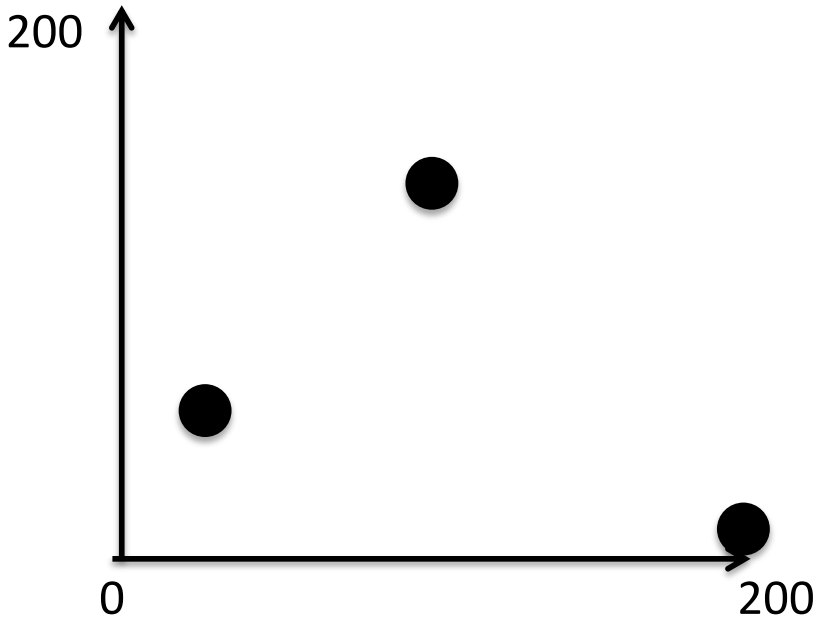
```
canvas.append("circle").attr("r",20);
canvas.append("circle").attr("r",20);

array1 = [5]
canvas.selectAll("circle")
        .data(array1)
        .exit()
        .remove()
```

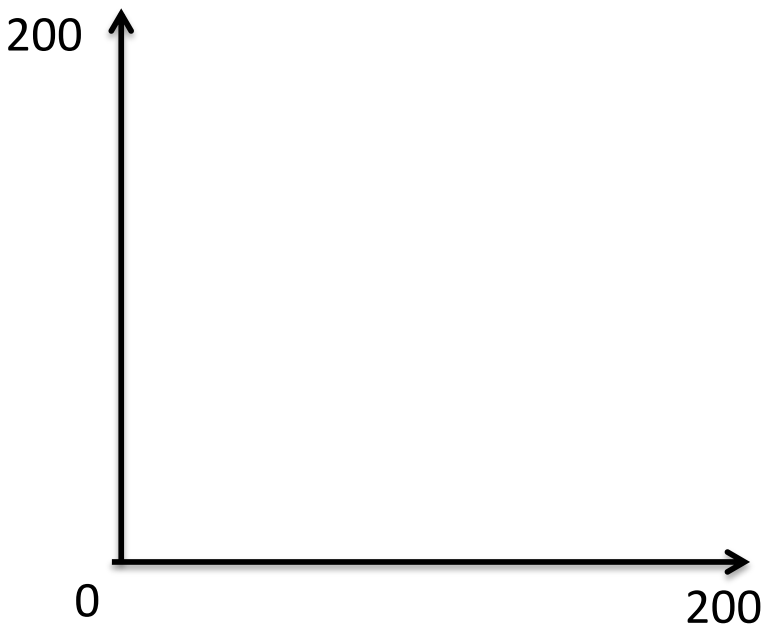
b) How many now circles will be left on the screen _____?

```
canvas.append("circle").attr("r",20);

array1 = [5, 6, 2]
canvas.selectAll("circlesssss")
        .data(array1)
        .exit()
        .remove()
```

6. Mark positions off the circles in the empty graph after executing the new code

```
canvas.append(“circle”).attr(“r”,20).attr(“cx”, 200) .attr(“cy”, 5);
canvas.append(“circle”).attr(“r”,20) .attr(“cx”, 20) .attr(“cy”, 50)
canvas.append(“circle”).attr(“r”,20) .attr(“cx”, 90) .attr(“cy”, 150)

// new code
array1 = [5, 2, 1]
canvas.selectAll(“circle”)
        .data(array1)
        .attr(“cx”, 0)
        .attr(“cy”, 0)
```





The 4th Homework

Possible STOPS

- Data points to Pixels
- Combining Arrays
- Years in the X axis
- How to draw lines

Algorithm draw scatter / line plot we have Year Beer arrays

1. Create a canvas
2. Add group
3. Give margins to canvas
4. Combine two arrays to one
5. Create Scales
6. Draw Axis
7. Convert data to pixels
8. Circles
9. Lines

Remove Shapes

First selection must be done of what to remove.

- If you select elements that doesn’t exist - there will be nothing at hand to remove
    - E.g. canvas.selectAll(“circlesss”) - no circlesss elements can be created
- Possible to use classes or ids as attributes to group shapes
    - E.g. canvas.append(“circle”).attr(“class”,”circlessss”)

      canvas.selectAll(“.circlessss”) will select all shapes with the same class no matter what type shape it is

Then we can remove either completely

```
selection.remove()
```

Or partially based on the length of array

```
selection.data().exit().remove()
```

We can also hide instead of removing objects

- selection.style(“display”, “none”)

And make them appear again

- selection.style(“display”, “block”)

.style() is a call to CSS – elements that are used for web pages styling

In this way we don’t need to redraw shapes, we can just hide them. But we are loosing possibility to animate movements

Update Axis

Update scale based on which axis is created

```
scaleX
    .domain([Year[0],Year[17-5]])
```

Select axis

needs identification assignment to axis in order to locate it

```
canvas.append("g")
    .attr("transform","translate(0,600)")
    .style("color","grey")
    .attr("class","xaxis")
        .call(xAxis)
```

Then we can select axis by class name, and call the function to redraw axis

```
canvas.select(".xaxis").call(xAxis)
```

- Numbers on the X axis might not make sense as it should be years, thus before redrawing axis we could convert them to date

```
var mindate = new Date(1998);
        var maxdate = new Date(2012);

scaleX
            .domain([mindate, maxdate])
            //.domain([Year[0],Year[17-5]])
```

Homework 5

- Update Homework 4 with additional lines and circles for Wine dataArrayt
- Include zoom buttons (+ and -) for X axis

Algorithm

- Draw canvas, circles, and lines
- We draw x Axis
  - Updating Scale
  - Updating Function for xAxis
  - Redrawing xAxis on canvas
- We adjust graph to Axis