Lab 4

Submit your team number

Question *Submitted May 3rd 2023 at 12:32:05 pm*

Please enter your team number.

12

1. Analyze the Data Using D3 Library Methods (Lab 3 Recap)

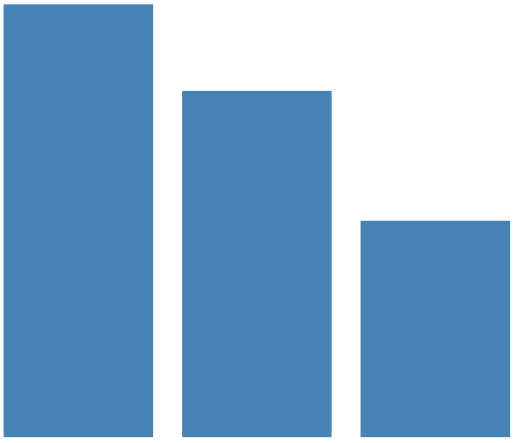
Download lab4.html and cereal.csv. The starter code loads cereal.csv, the same dataset you used in Lecture Exercise 5; and generates a <svg>

This time, draw a barplot with the number of cereals per Manufacturers.

d3.rollup() will help you calculate the number of cereals per Manufacturers.

Remember **d3.rollup() returns an `InternMap` object,** not an `array` like `data`. When defining anonymous functions (i.e. `function(d) {...}`) for the attributes of `<rect>`, checking what `d` looks like by using `console.log(d)` will help you correctly define those anonymous functions.

Your output should look similar to this below:



Question *Submitted May 3rd 2023 at 12:56:54 pm*

Copy paste the code you added.

```
const cerealCounts = d3.rollup(data, v => v.length, d => d.Manufacturer)

// drawing the barplot, without dynamic scaling for now (slide 1)
svg.selectAll(".bar")
    .data(cerealCounts).enter()
        .append("rect")
    .attr("class", "bar")
    .attr("x", (d, i) => i * 5)
    .attr("y", d => height - d[1])
    .attr("height", d => d[1])
    .attr("width", d => 3)
    .attr("fill", "blue")
```

2. Dynamic Scaling

Now let's add dynamic scaling.

The number of cereals is a quantitative attribute so d3.scaleLinear() works. For categorical/ordinal attributes, d3.scaleBand() is a better choice. With bandwidth() you can also set the width each bar dynamically.

Take a look at the hyperlink above and integrate dynamic scaling for both x- and y-axis on the barplot.

Question 1 *Submitted May 3rd 2023 at 1:30:10 pm*

Copy paste your xScale and yScale

```
let xScale = d3.scaleBand()
    .domain(dataArray.map(d => d.key))
    .range([0, width])
    .padding(0.1)
let yScale = d3.scaleLinear()
    .domain([0, d3.max(dataArray.map(d => d.value))])
    .range([height, 0])
```

Question 2 *Submitted May 3rd 2023 at 1:30:01 pm*

Copy and paste your modified code to generate the barplot with dynamic scaling.

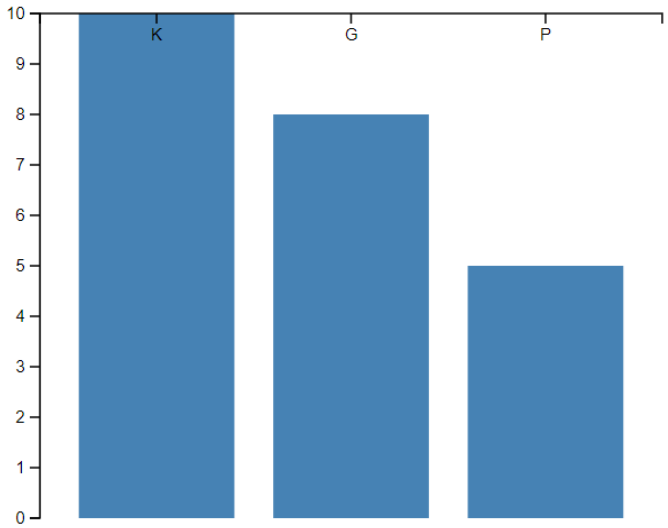
```
svg.selectAll(".bar")
    .data(dataArray).enter().append("rect")
    .attr("class", "bar")
    .attr("x", d => xScale(d.key))
    .attr("y", d => yScale(d.value))
    .attr("width", xScale.bandwidth())
    .attr("height", d => height - yScale(d.value))
    .attr("fill", "blue");
```

3. Add Axis to the Cereal barchart

Now we would like to add the X and Y axis to the barchart. Follow the instructions below.

- Define xAxis and yAxis variables. You can find the full list of axis methods here. You can refer to the lecture slides today too.
- Adapt xAxis and yAxis to the current barchart.

<Resulting barchart without transformation>



After adding the axes, your barchart should be similar to this (color does not matter).

Question 1 *Submitted May 3rd 2023 at 1:34:20 pm*

As discussed in the lecture, you will notice that the output scatter plot doesn't look right. "transform" attribute is key to resolve the two issues we discussed in the lecture. Take a look at the demo of each transformation in this article and answer the following questions.

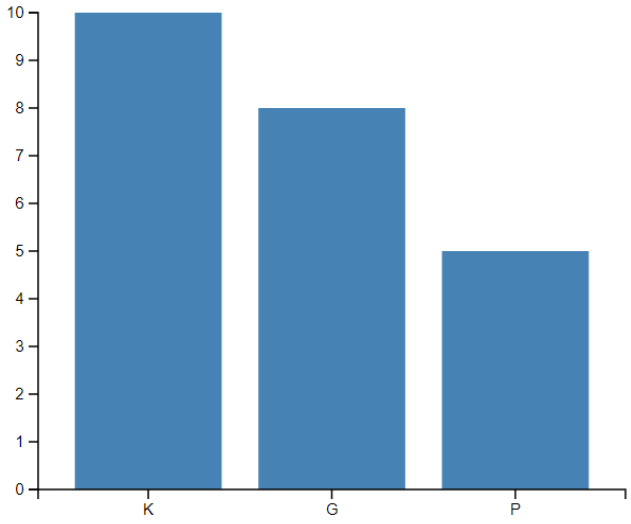
What transformation do we need to resolve the issues about the cereal scatterplot?

```
svg.append("g").call(d3.axisBottom(xScale))
    .attr("class", "xAxis")
    .attr("transform", `translate(0, ${height})`)
svg.append("g").call(d3.axisLeft(yScale))
    .attr("class", "yAxis")
```

Question 2 *Submitted May 3rd 2023 at 1:34:42 pm*

Now refer to the article again and check out the d3 implementation of the transformation of your choice. **Copy and paste your modified implementation (call) of the axis.**

Your modified barchart should look like this:

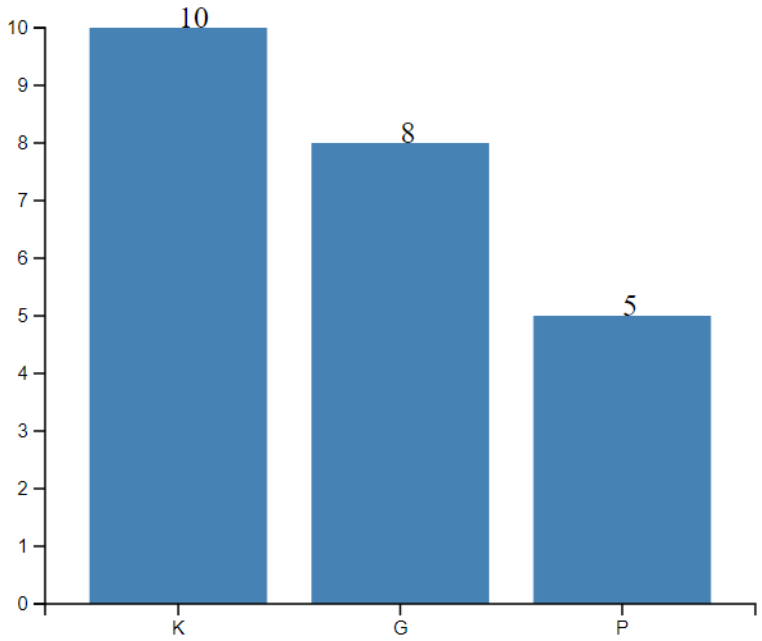



```
        svg.append("g").call(d3.axisBottom(xScale))
            .attr("class", "xAxis")
            .attr("transform", `translate(0, ${height})`)
        svg.append("g").call(d3.axisLeft(yScale))
            .attr("class", "yAxis")
```

4. Add Labels to the Barchart

Add labels of number of cereals for each manufacturer, on the top of each bar with font size "16px".

Your output should be similar to:



Question *Submitted May 3rd 2023 at 1:36:49 pm*

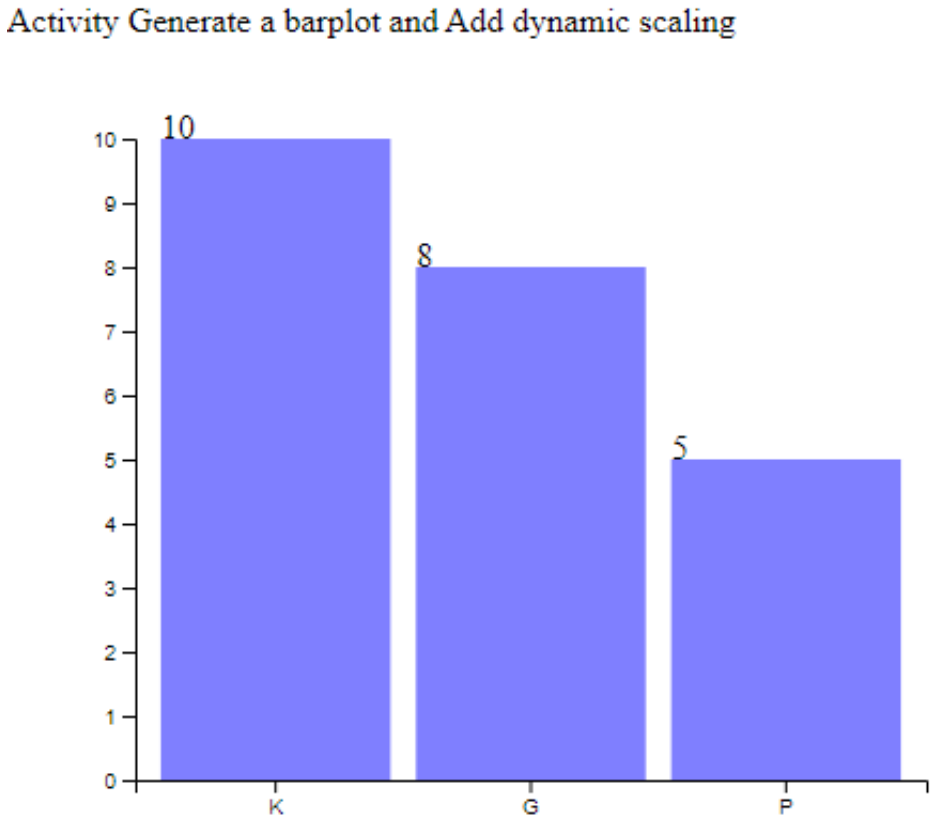
Copy and paste the part of your code which adds labels of cereal names.

```
svg.selectAll(".bar-labels")
    .data(dataArray).enter().append("text")
    .attr("x", d => xScale(d.key))
    .attr("y", d => yScale(d.value))
    .attr("text-anchor", "right")
    .text(d => d.value)
```

Upload Your Files

Question 1 *Submitted May 3rd 2023 at 1:37:02 pm*

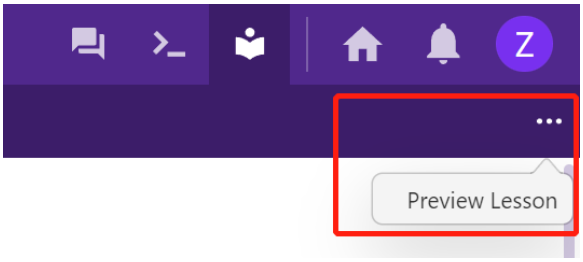
Upload the screenshot of your resulting webpage. You will need to click the "clip" button to upload a file into the Answer box.



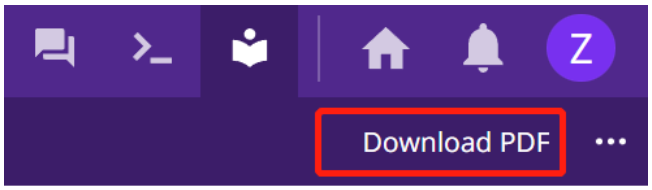
Question 2

You need to download the PDF of lecture exercise 3 and upload it with other files to the Gradescope. Follow the instructions on how to download PDF file:

1. Click on the ellipsis button and the Preview Lesson.



2. After that, click on the Download PDF button.



- [ ] PDF downloaded!

- [ ] Haven't done yet!

Question 3

Upload the following files to Gradescope. You need to make **a group submission,adding all present members in your team**, so that the present members get the participation credit.

Files to upload:

- lab4.html
- PDF you downloaded as Q2

- [ ] Our team uploaded the the files on gradescope!

- [ ] Oops, our team did not upload the files on gradescope!

Feedback

Question

Was the activity today clear? If not, please share how the course can improve it. Your comments will help us design future lab content (and also future students).

No response