

Lab 3

Submit your team number

Question *Submitted Apr 26th 2023 at 2:04:57 pm*

Please enter your team number.

1. Write your callback function to convert all numeric columns into float/int

Now that you have an understanding how to debug, let's get started with our analysis of [cereal.csv](#) in [lab3.html](#). **Download** the two files and make sure they are **in the same folder**.

First we will load `cereal.csv` to `lab3.html`, using `d3.csv()`. The second argument of `d3.csv()` is a [callback function](#). The callback function allows us to do any data processing work, while loading the csv file (convenient!).

`Lab3.html` already defined an empty callback function `rowConverter()`. This function takes a row of the csv file as an input, which is represented as an array of Strings, and returns an object which has a list of key-values. Notice that the Calories, Carbo, and Year have numerical data yet they are going to be represented as Strings when `d3.csv()` is called; so `rowConverter()` should convert these String values to either integer or float. To do so, you can use `parseInt()` and `parseFloat()` functions.

cereal.csv					
	A	B	C	D	E
1	Name	Manufacturer	Calories	Carbo	Year
2	All-Bran	K	70	7	2015
3	Apple Cinnamon Ch	G	110	10.5	2016
4	Bran Flakes	P	90	13	2019
5	Cheerios	G	110	17	2020
6	Cinnamon Toast Cru	G	120	13	2015

Complete `rowConverter()` on `lab3.html` so that the Name and Manufacturer keys hold String values while the Calories, Carbo, and Year keys hold integer/float values. Following example will help you construct your `rowConverter()`.

If we have a CSV file like:

`cities.csv:`

```
city,state,population,land area
seattle,WA,652405,83.9
new york,NY,8405837,302.6
boston,MA,645966,48.3
kansas city,MO,467007,315.0
```

We can write a callback function `rowConverter` like this.

```
var rowConverter= function(d){
```

```
return {
  city : d.city,          // key: city, value: d.city
  state : d.state,        // key: state, value: d.state
  population : parseInt(d.population), // key: population, value: int of d.population
  land_area : parseFloat(d["land area"]) // key: land_area, value: float of d["land area"]
}
}
```

Question 1 *Submitted Apr 26th 2023 at 2:12:32 pm*

Have you written your own **rowConverter** function which can convert all numeric columns of [cereal.csv](#) into int/float?

☒ Yes, we are done!

☐ No we are not done

Question 2 *Submitted Apr 26th 2023 at 2:12:55 pm*

Copy/paste your own rowConverter here which can convert all numeric columns of [cereal.csv](#) into int/float.

```
let rowConverter = function(d){
  return {
    name : d.Name,
    manufacturer : d.Manufacturer,
    calories : +d.Calories,
    carbo : +d.Carbo,
    year : +d.Year
  }
}
```

2. Analyze the data using D3 library methods

You can read a csv file, say [cereal.csv](#), by using `d3.csv()`. To do so, you need to [load](#) and parse data using `rowConverter()` as a callback function.

For example, if we want to load and parse a csv file "cities.csv" with our `rowConverter()` and output the data to the console, we can use the following code:

```
d3.csv("cities.csv", rowConverter).then(function(data) {  
  console.log(data);  
  
  // this is where you add the code to do the tasks below  
  
});
```



Make sure your added code blocks are **inside** the curly bracket.

Question 1 *Submitted Apr 26th 2023 at 2:17:26 pm*

First of all, you need to Group the cereals by manufacturers. To do this, we need to use [d3.group\(\)](#). Also, using `console.log()` to output your result.

Copy/paste the relevant part of your code.

```
let manufacturers = d3.group(data, d => d.manufacturer)  
console.log(manufacturers)
```

Question 2 *Submitted Apr 26th 2023 at 2:21:35 pm*

Then, get the sum of carbo, per manufacturer by using [d3.sum\(\)](#) and [d3.rollup\(\)](#). Also, using `console.log()` to output your result.

Copy/paste the relevant part of your code.

```
let sums = d3.rollup(data, v => d3.sum(v, d => d.carbo ), d => d.manufacturer)  
console.log(sums)
```

Question 3 *Submitted Apr 26th 2023 at 2:28:26 pm*

Finally, find the sum of calories for manufacturer "K" in 2020 by [d3.sum\(\)](#) and [.filter\(\)](#). Using

`console.log()` to output your result.

Copy/paste the relevant part of your code.

```
let sumK = d3.sum(data.filter((d, i) => d.manufacturer === 'K' && d.year === 2020),
console.log(sumK)
```

Question 4 *Submitted Apr 26th 2023 at 3:03:18 pm*

Now, we would like to draw a barchart to illustrate the sum of calories for different manufacturers as we done in Q2. Your barchart should look similar to this:



- The first bar is the summation of Carbo of Manufacturer "K". The second one is for "G" and the third one is for "P".
- The color doesn't matter.

Hint: You may need to think how to normalize the width and height of each bar with the given `width` and `height` of the svg element.



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.

Copy/paste your code below.

```
let t = [...sums.keys()]
let maxVal = 0
let length = 0
for (let i in t) {
  const key = t[i]
  const val = sums.get(key)
  if (val > maxVal) { maxVal = val }
  length = +i + 1
}
```

```
const xScale = width / maxVal
const yScale = height / length
svg.selectAll('.bar')
  .data(sums).enter()
  .append("rect")
  .attr("class", "bar")
  .attr("width", d => d[1] * xScale)
  .attr("height", yScale - 2)
  .attr("x", 0)
  .attr("y", (d, i) => i * yScale)
  .attr("fill", "#69b3a2")
```

Question 5 *Submitted Apr 26th 2023 at 3:03:46 pm*

Are you done with the activity and ready to submit?

☒ Yes, we are done!

☐ No we are not done

Upload Your Files

Question 1 *Submitted Apr 26th 2023 at 3:04:00 pm*

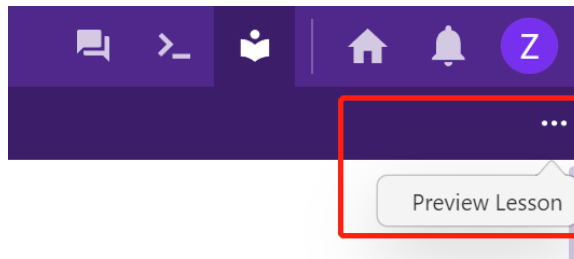
Upload the screenshot of your resulting webpage (Your console information). You will need to click the "clip" button to upload a file into the Answer box.



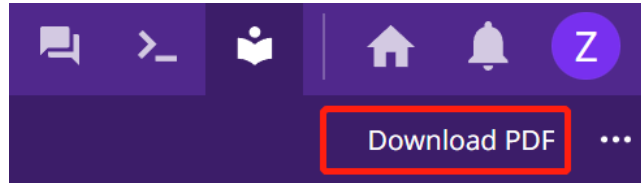
Question 2

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:

- lab3.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