Javascript D3 Histogram: thresholds producing wrong number of bins

Asked 3 years ago Active 1 year, 10 months ago Viewed 2k times

2

I'm in the process of creating a histogram JS script using D3, and it all seems to be working correctly... except for the number of bins.

Following is the relevant part of my code:

```
//Define the scales for the x and y attributes
var x = d3.scaleBand()
    .range([0, width])
    .padding(configProperties.barPadding);
var y = d3.scaleLinear()
    .range([height,0]);

//Create the bins
var bins = d3.histogram()
    .domain(d3.extent(data))
    .thresholds(configProperties.binsCount)
    (data);

console.log("number of bins: " + bins.length); //9
console.log("intended number of bins: " + configProperties.binsCount); //10
```

If I set configProperties.binsCount to 9, bins.length is still 9. If I set configProperties.binsCount to 14, bins.length is still 9.

If I set binsCount to 15 or higher, however... bins.length outputs 23.

My understanding of how histogram.thresholds works based on the documentation is that if I give it a value, it will divide the data into that many + 1 equal segments (i.e. that many bins). However, it doesn't seem to be doing that at all. All of the example code that I could find seemed to indicate that I am using it correctly, but I can't get the number of bins that I need.

I've also tried using d3.ticks as a thresholds argument, but I encounter the same issue.

Is there something I'm missing? Does it have to do with my domain? Thanks in advance.

javascript d3.js histogram data-visualization

asked Apr 12 '17 at 22:28
user7859002
43 6

2 Answers

Active Oldest Votes

3

You are passing a **count** (that is, a simple number) to the `thresholds` function, **not** an array.

What you're seeing is the expected behaviour when you pass a number. According to the same docs:

> If a count is specified instead of an array of thresholds, then the domain will be uniformly divided into approximately count bins;

Let's see it in this demo:

```
var data = d3.range(100);

const histogram = d3.histogram()
  .value(d => d)
  .thresholds(5);

var bins = histogram(data);

console.log("The number of bins is " + bins.length)


<script src="https://d3js.org/d3.v4.js"></script>
```

Run code snippet Copy snippet to answer Hide results Full page

```
The number of bins is 4                              00:33:20.986
```

As you can see, `count` is 5 and the number of bins is also 5.

If you pass an **array**, however, the behaviour is what you expect: the number of bins will be *array.length* + 1:

> Thresholds are defined as an array of values [x0, x1, …]. Any value less than x0 will be placed in the first bin; any value greater than or equal to x0 but less than x1 will be placed in the second bin; and so on. Thus, the generated histogram will have thresholds.length + 1 bins.

Here is the demo:

```
var data = d3.range(100);

const histogram = d3.histogram()
  .value(d => d)
  .thresholds([10, 30, 50, 70, 90]);

var bins = histogram(data);

console.log("The number of bins is " + bins.length)


<script src="https://d3js.org/d3.v4.js"></script>
```

Run code snippet   Copy snippet to answer   Hide results   Full page

```
The number of bins is 6          00:44:28.431
```

As you can see, the array has 5 values and the number of bins is 6.

Finally, have in mind that the actual number of bins **depends on the data** you pass to the histogram generator. That explains the other results you're describing in your question.

edited Apr 13 '17 at 3:40   answered Apr 13 '17 at 2:43
Gerardo Furtado
81.2k 9 68 120

Thank you for the response! So, I suppose the real answer as to why I'm not getting the expected number of bins is because my dataset disallows it? I find that a bit confusing, because if I, for example, use an online histogram plot maker, it has no problem charting the data using any number of bins (i.e., 10 as per my example.) Here is a screenshot showing what I mean. – user7859002 Apr 13 '17 at 18:18

I think I figured out how to do this by manually calculating my own *array* for thresholds based on the data and provided as well as binsCount. This seems to work better than using *count* . Thanks again for helping me think through this. – user7859002 Apr 13 '17 at 18:44

3

I realize this is a little old, and that Gerardo explained how to do what you were asking, but he didn't actually answer the *why* of the question. So here's that, in case anyone else comes across this question and is curious. If you pass a number to the thresholds function, D3 finds a number of bins that is near to that number, such that the thresholds are 'nice' numbers. And it's the choosing of those 'nice' numbers that results in the number of bins being different than what you specify.

So if your data goes from 0 to 24.37, and you request 8 bins, the thresholds will not be multiples of 3.481428571428... (= 24.37 / (8-1)). Instead D3 will pick a 'nice' maximum of 25, and the threshold will be multiples of 2.5 (to make 10 bins) or multiples of 5 (to make 5 bins). These numbers are much nicer to display on a graph, and are what a human would probably choose if they were making the histogram by hand.

answered Jun 6 '18 at 15:12

DaveTheScientist
2,561 18 19