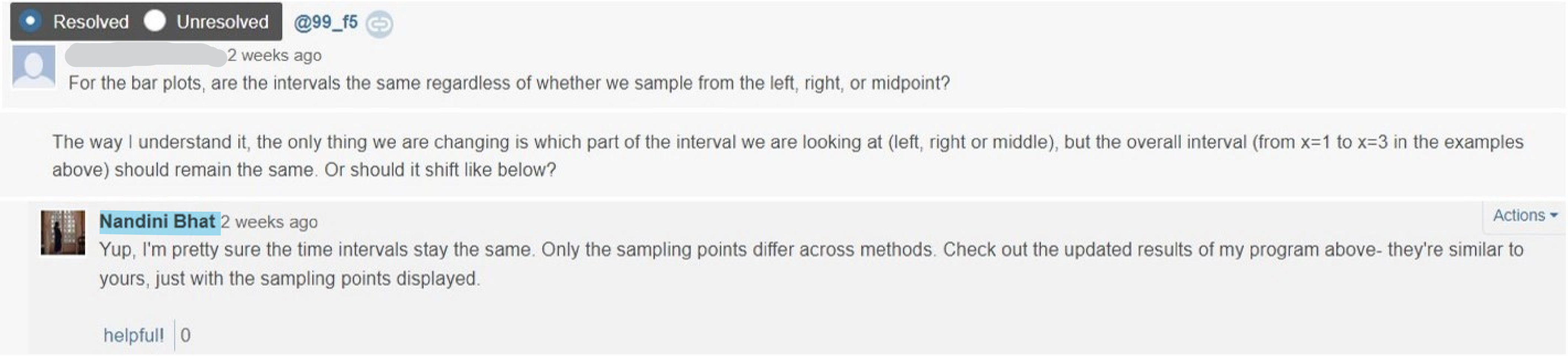


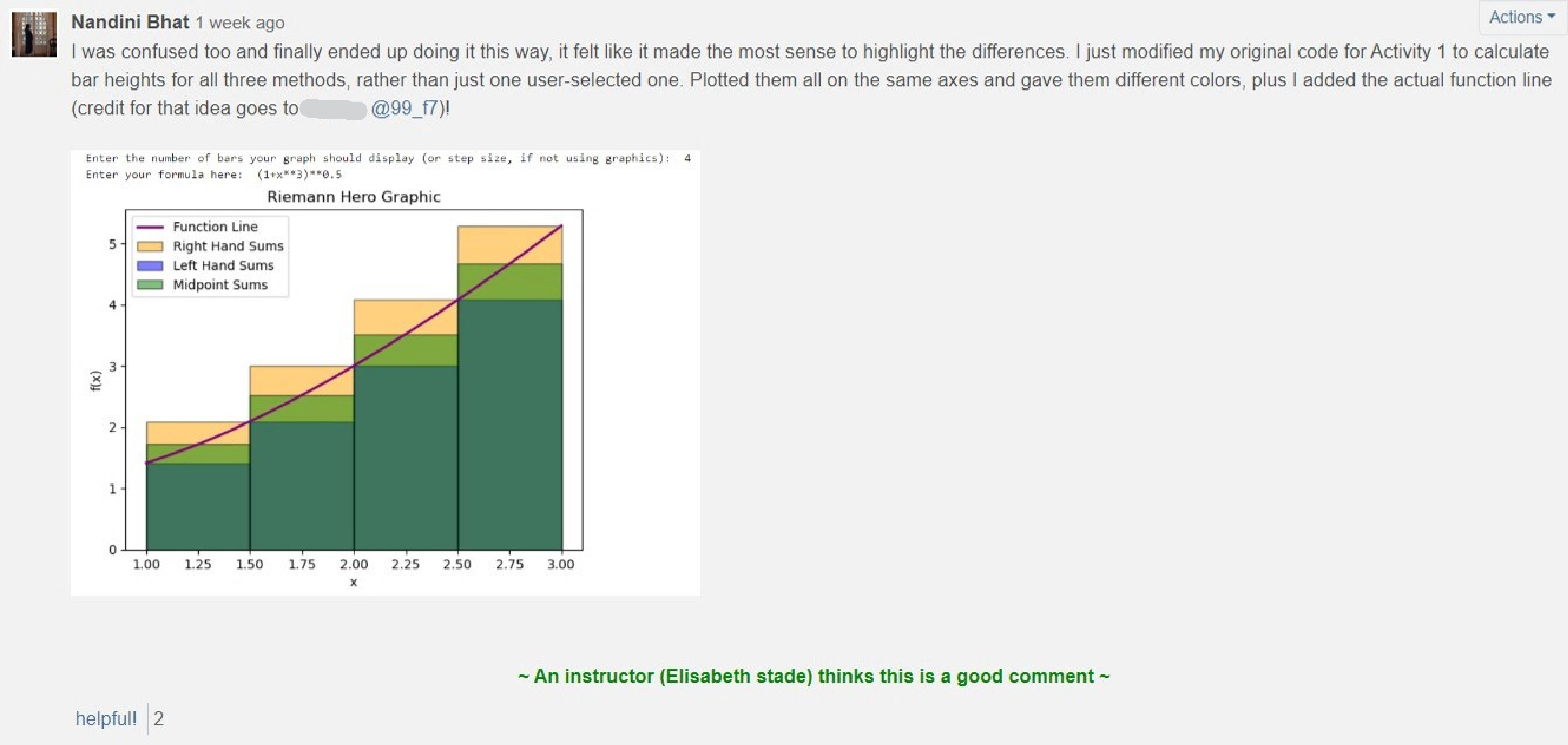
[redacted] 2 weeks ago

For the bar plots, are the intervals the same regardless of whether we sample from the left, right, or midpoint?

The way I understand it, the only thing we are changing is which part of the interval we are looking at (left, right or middle), but the overall interval (from x=1 to x=3 in the examples above) should remain the same. Or should it shift like below?

Nandini Bhat 2 weeks ago

Yup, I'm pretty sure the time intervals stay the same. Only the sampling points differ across methods. Check out the updated results of my program above- they're similar to yours, just with the sampling points displayed.



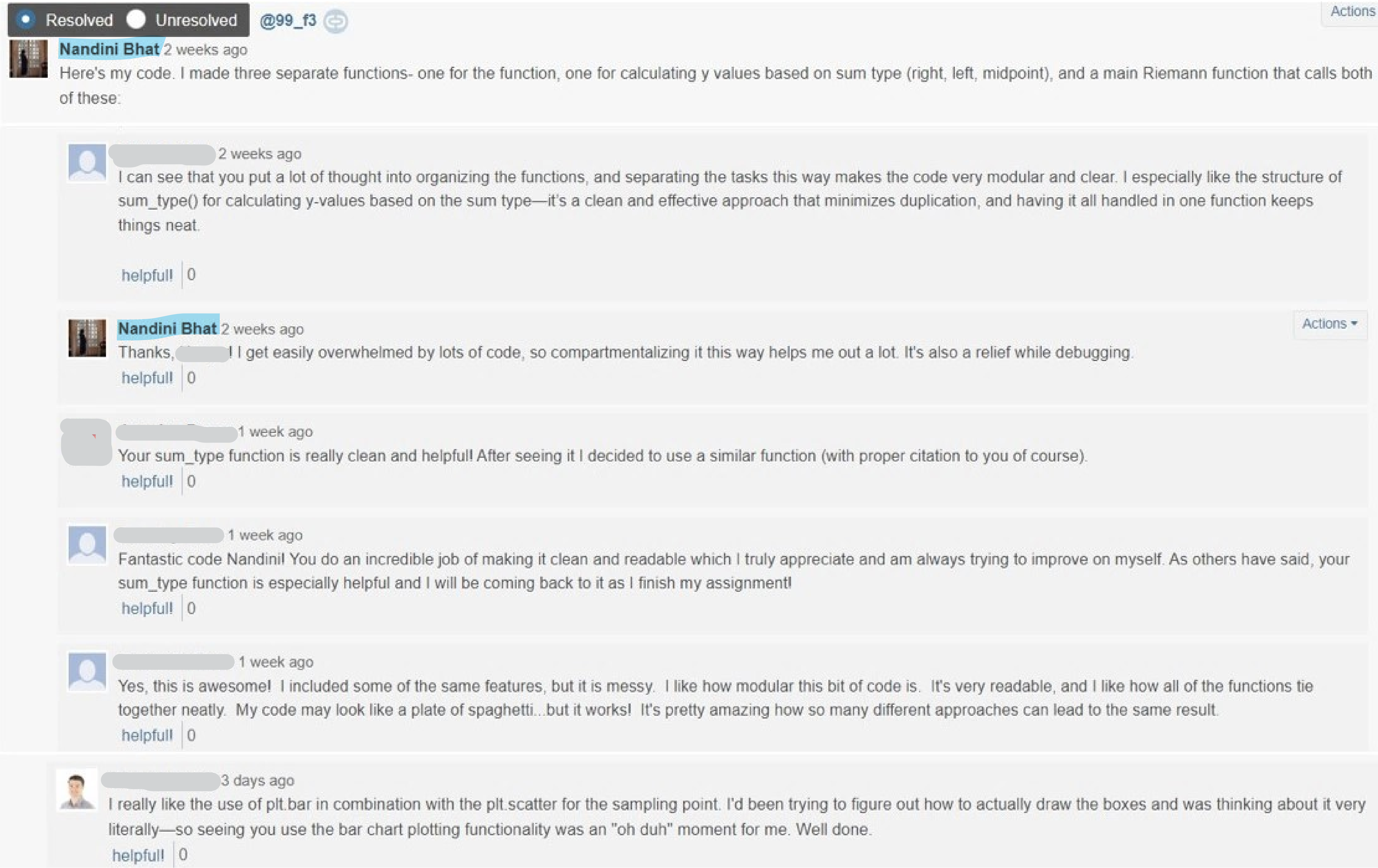
Nandini Bhat 1 week ago

I was confused too and finally ended up doing it this way, it felt like it made the most sense to highlight the differences. I just modified my original code for Activity 1 to calculate bar heights for all three methods, rather than just one user-selected one. Plotted them all on the same axes and gave them different colors, plus I added the actual function line (credit for that idea goes to [redacted] @99_f7)!

```
Enter the number of bars your graph should display (or step size, if not using graphics):  4
Enter your formula here:  (1+x**3)**0.5
```

Riemann Hero Graphic

~ An instructor (Elisabeth stade) thinks this is a good comment ~



Nandini Bhat 2 weeks ago

Here's my code. I made three separate functions- one for the function, one for calculating y values based on sum type (right, left, midpoint), and a main Riemann function that calls both of these:

[redacted] 2 weeks ago

I can see that you put a lot of thought into organizing the functions, and separating the tasks this way makes the code very modular and clear. I especially like the structure of sum_type() for calculating y-values based on the sum type—it's a clean and effective approach that minimizes duplication, and having it all handled in one function keeps things neat.

Nandini Bhat 2 weeks ago

Thanks, [redacted]! I get easily overwhelmed by lots of code, so compartmentalizing it this way helps me out a lot. It's also a relief while debugging.

[redacted] 1 week ago

Your sum_type function is really clean and helpful! After seeing it I decided to use a similar function (with proper citation to you of course).

[redacted] 1 week ago

Fantastic code Nandini! You do an incredible job of making it clean and readable which I truly appreciate and am always trying to improve on myself. As others have said, your sum_type function is especially helpful and I will be coming back to it as I finish my assignment!

[redacted] 1 week ago

Yes, this is awesome! I included some of the same features, but it is messy. I like how modular this bit of code is. It's very readable, and I like how all of the functions tie together neatly. My code may look like a plate of spaghetti...but it works! It's pretty amazing how so many different approaches can lead to the same result.

[redacted] 3 days ago

I really like the use of plt.bar in combination with the plt.scatter for the sampling point. I'd been trying to figure out how to actually draw the boxes and was thinking about it very literally—so seeing you use the bar chart plotting functionality was an "oh duh" moment for me. Well done.