Feedback for Emma Husby, Charae Gilbert, Sam Borghese, Evan Leu, Nick Rezaee, Matthew Rhea

The introduction is concise and pretty effective. But when you say "two important observations," the things you describe are confusing to me. I think I know what you mean by "Gaussian density of primes near a given number N." But what is the *observation* you refer to when you write "prime gaps between spaces of consecutive primes." I would expect an observation *about* prime gaps. Otherwise, you're describing *what* you're observing and not what you observed. E.g., "two important observations about animals: lions and zebras" sounds weird.

The code looks good up through the list of primes. I worry a bit about the `prime_diffs_list` function, because as you write it, you are calling the `isprime_list(n)` function twice. This could double the time taken, though Python is sometimes smart enough to figure out that you really only need the function result once. The best way to fix this would be to add a line like `plist = where(isprime_list(n))` before the return, and then use `plist` twice in the return statement.

The rest of the code looks pretty good, though list slicing and broadcasting would be better in places. For example, instead of `np.array([np.log(x+1)+1 for x in x_values])`, you can write `np.log(x_values+1)+1`. (It will probably run faster).

You produce two plots to study prime gaps. I like the idea of plotting a "running average" and comparing the data to this running average. Comparing the running average to the log is also a great idea, given what Gauss knew about the density of primes. The plots could use some more refinement, along the following lines:

- When you're overlaying multiple plots, it's best to display a *legend* to describe what's what. You try to do this on the y-axis label, but it's not too clear.
- The x-axis could simply be labeled "prime numbers" and the y-axis "gap (distance to next prime)". The legend can describe what's the actual data, what's the rolling average, and what's the expectation (log).
- For discrete data like this, I think the blue line graph should be replaced by a scatterplot. Otherwise, there's a lot of clutter in the image.
- The y-axis ticks should probably be located at 1, 2, 4, 6, 8, etc., since these are the only y-values that occur. The 5-unit grid isn't as helpful. Similarly, you could – if you wanted – experiment with changing the x-axis ticks to actual prime numbers.

I'm a little more confused about the second matplotlib graph – the list-constructions could be simplified a lot, and the code isn't documented well enough for me to figure out what your goal is. You should give some markdown cells betwen code blocks to explain what's going on. I think that your second graph is comparing the n^{th} prime number with the number $n \log(n)$, roughly speaking. I found the construction a bit confusing, but the closeness of these numbers is displayed nicely in the closeness of the blue and orange graphs. Again, a legend and better axis titles would be helpful.

Update: I took a look at the PDF file emailed afterwards, which improves the plots significantly (with legends, etc.). Still, I'm a bit confused by the graphs – there needs to be some explanation of what is being portrayed in each, and which I should focus on. The goal was really to produce *two* key visualizations, and there are a lot of them here without a natural flow.

The PIL visualization is much clearer to me – the code is clean and well-commented. The visualization is beautiful and fascinating. It's striking how much information is visible. A side-effect of your visualization technique is that prime gaps are displayed in black along the diagonal – larger gaps as larger black squares. The squares propagate into the blue and red cross-hatched pattern, as a long string of small gaps corresponds to an "excess" of primes large gaps correspond to a "deficiency" of primes. It's just remarkable... and I haven't seen a visualization like it. Maybe it's lucky... but this kind of "let's see what happens" is the source of beautiful patterns. There are a few modifications I might try – one is that the error `errorarray[x,y]` should be compared to $\sqrt{|y-x|}$, due to the expectations from the Riemann hypothesis. I'll have to see what pattern comes out of this, rather than the relative error that you use for coloring.

I'm (slowly) making a coffee-table style book about prime numbers. With your permission, I'll produce an image like this one to include, and acknowledge that the ideas were originally due to your group.

Grade breakdown.

Coding tasks: 9/10
Quality of visualization: 4/5
Depth of exploration: 5/5
Total: 18/20.