

Summary: The project I reviewed was Larry's, Ruth's, and Uzodinma's project. They proposed to create a software engineer salary geographic visualization that would allow for users to view and compare the salaries of software engineers across the global. The granularity of the geographic visualization is on countries, wherein average salaries, education/training level, and average ages are the statistics each country selection will provide. They are using data retrieved from the 2015 StackOverflow Developer Survey.

Strengths: Below are the strengths I found that their proposal included.

1. **Simplicity.** Their visualization is designed to answer simple questions and present information as simply as possible. Simplicity in this instance is wonderful being deliberate on their predictions and permits an easy interface for which the user to interact with.
2. **Lightweight.** The computations required for this visualization are only percentages and averages, allowing for a quick and expendible visualization that will more easily run across aging computer platforms.
3. **Abundance of information.** The dataset itself is easily accessible but can be hard to read for an untrained eye. This visualization provides an easy medium with which users can peek into the various statistics they want to display.

Weaknesses: Below are the weaknesses I found that their proposal included.

1. **Is a visualization necessary?** Moreover, this data might be more easily displayed as a table or a series of bar and pie charts. Although a geographic visualization is interesting, the visualization not only undermines the area principle (some countries might also be hard to find like Luxembourg) but also overcomplicating rather simple questions that might be more easily and cleanly answered with simpler charts.
2. **Salary conversions and PPP.** There is no mention in the proposal about what a salary means and how one salary in one country compares to another. This lack of information is extremely problematic if comparisons are being drawn because if a software developers in France and India make the same income (say conversions all work

out and the salaries are equal in USD), the software developers in India have much higher purchasing power parity (PPP) than their French counterparts. This issue effectively undermines any comparisons one can create with regards to salaries unless the dataset itself adjusted for this issue.

3. **Assumes all users have equal access to resources.** Another problematic issue is the question of user education/training. What's not clear from the data is if fewer people in country A are going to college than country B, is it because country A has cultural issues with certain population's attending higher education or do they have structural barriers that are preventing people from attending college. Therefore I don't believe comparing educations is a useful metric or even thoughtful question.

General Comments: I believe that if you want to use a data visualization, specifically this geographic overlay coupled with the StackOverflow dataset, it'd be best to motivate the visualization with more complex questions, such as gender representation, age correlation with higher salary, higher education with salary, etc. Additionally, salaries across countries is difficult to capture since not all programmers are on StackOverflow, and even fewer might make a salary. Therefore, it might be more useful to instead survey a particular country, which will in itself have a very rich and diverse software engineering market, to get a better sense of the internal structure and benefits of software developers in a more closed environment. Effectively what I'm saying is that country might be too broad of a metric to get an accurate picture of software developers. I do like the overall design because of its simplicity and I encourage to stick with it where possible. Of course like stated above, simplicity is best displayed through simple visualizations, so make your judgment call there. Good luck and I look forward to seeing how this turns out!