Final Project GCBV Plan

• How much effort on average do you estimate it will take to refactor a view to use GCBVs?

Given that my final project shares many structural similarities with the EZU project, I anticipate that the effort required to refactor the view using GCBVs will be moderate to low, as I can use EZU project and the GCBV tutorial as the references.

• Based upon the estimate above, what do you estimate it will take to refactor all of your views?

I think that when refactoring my views, the data story and data visualization functions may need to be considered the most, as these two sections include the most dynamic content. The remaining dimensions have less dynamic content, but they will still benefit from switching to generic views. Because generic views enable a dynamic display of the user's choices at every step.

Based on my analysis, I anticipate that the data story, data visualization, data dimension, and file upload pages in my final project, especially the add, modify, and delete functionalities, will be refactored using generic views. Conversely, for pages like argument, motivation, and attitude, where users cannot edit or delete dimensions, the focus will be on implementing effective edit and update views. Therefore, the edit and update functionalities are likely to be the most crucial in these cases.

• Do you have any views in your Final Project that do not lend themselves to using GCBVs? Why not?

In my final project, the file upload views might not be significantly affected by using GCBV, given its relatively fixed functionality. However, for pages where the easy demonstration of dynamic content is crucial, GCBV seems to be an excellent choice.

• How many lines of code do you estimate would be eliminated from your Final Project by refactoring to GCBVs?

Based on my refactoring experience with EZU, I will estimate that refactoring to GCBV will reduce approximately 100 lines of code for each function.