Current Live Prototype: https://www.devinlange.com/research/MTAT/dist/index.html
Working:
- Data Model
- Data loading from file
- Loading from example files
- Non-interactive Plot of all paths
- Framework for setting up view and adding different visualizations

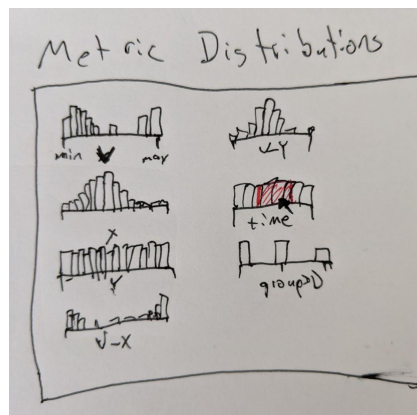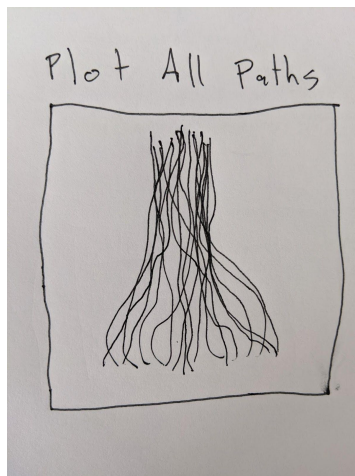My first round of design exploration involved a lot of ideation for different visualizations that could be applied to Multivariate Trajectory data:
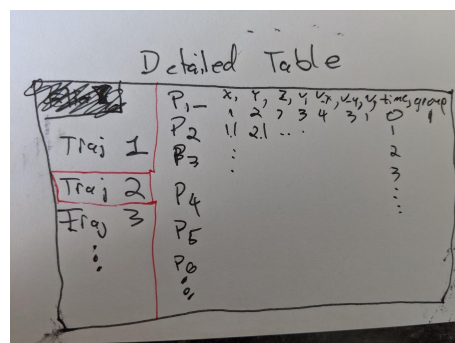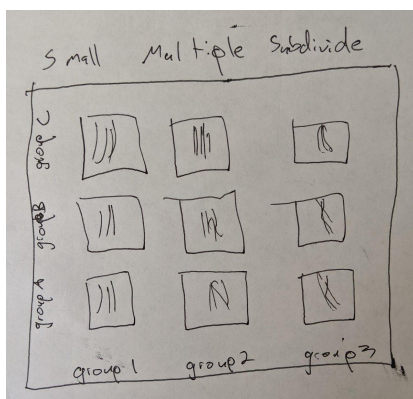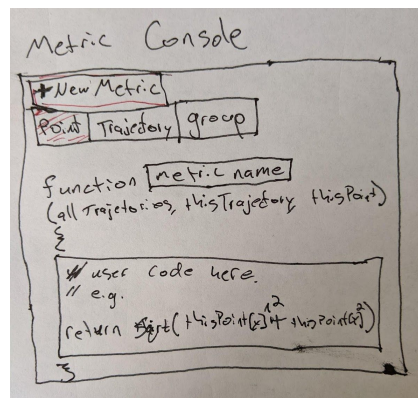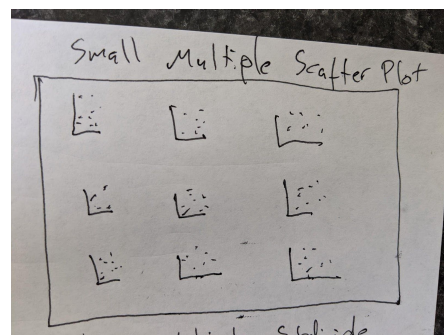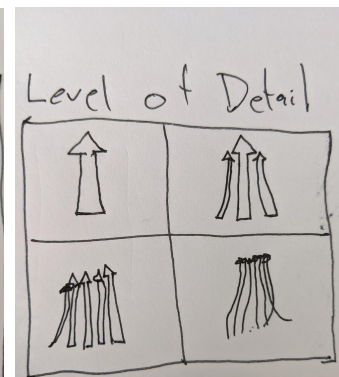
An early discussion about some of these yielded some new insights. First for the level of detail visualization, using a hierarchical clustering technique was another option (other than k-means clustering).

Another discussion was around the scatterplot matrix. I got the feedback that scatterplot matrices that show all combinations are often overkill, and not that useful. Instead it would be better to let the user select the combinations. When thinking through how to let users select which options, I did draw inspiration from the scatterplot matrix for how to let the user select different combinations quickly by giving a matrix they can select and deselect options.
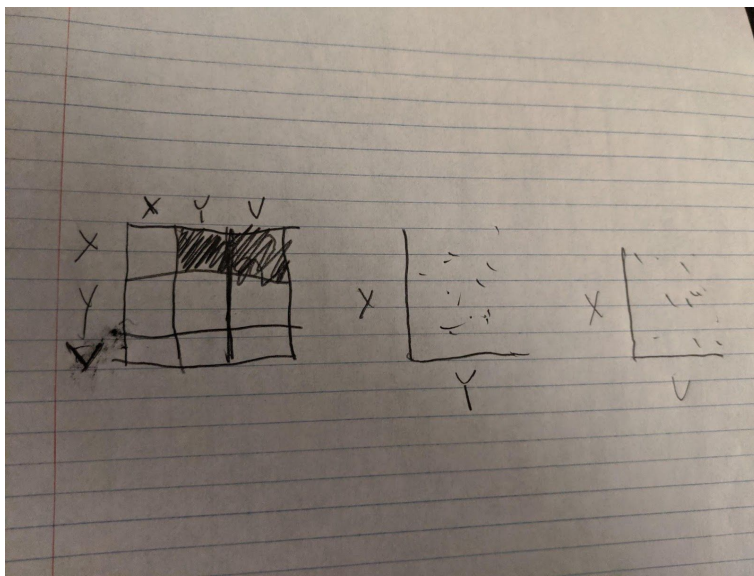

After I had this idea I was exposed to scatterplot matrices with distributions along the diagonal. For example:

From https://www.d3-graph-gallery.com/graph/correlogram_histo.html
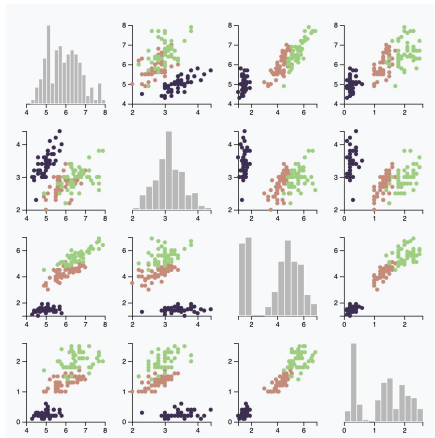
This concept clicked with my widget for selecting scatterplot options, and I decided that I would do something like this, in effect combining these two separate visualization ideas from before into one:
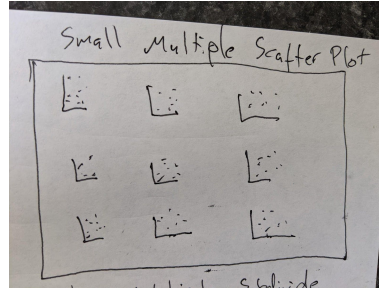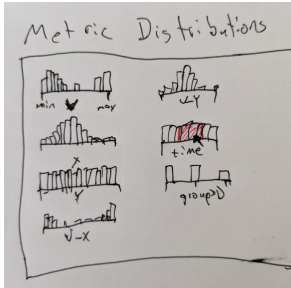

Another possible area of integration of multiple plots is within the tabular view of the data. One nice way of doing this that was pointed out during the visualization feecback session was with scented visualizations. Another separate topic that was discussed at these meetings was the mapping of variables onto the trajectories with color mappings.

As I was writing the code for the layout framework I realized I never combined these sketches into a cohesive layout. I did write the code so that it should be relatively easy to rearrange the layout, but I made an initial layout for the first set of visualization.
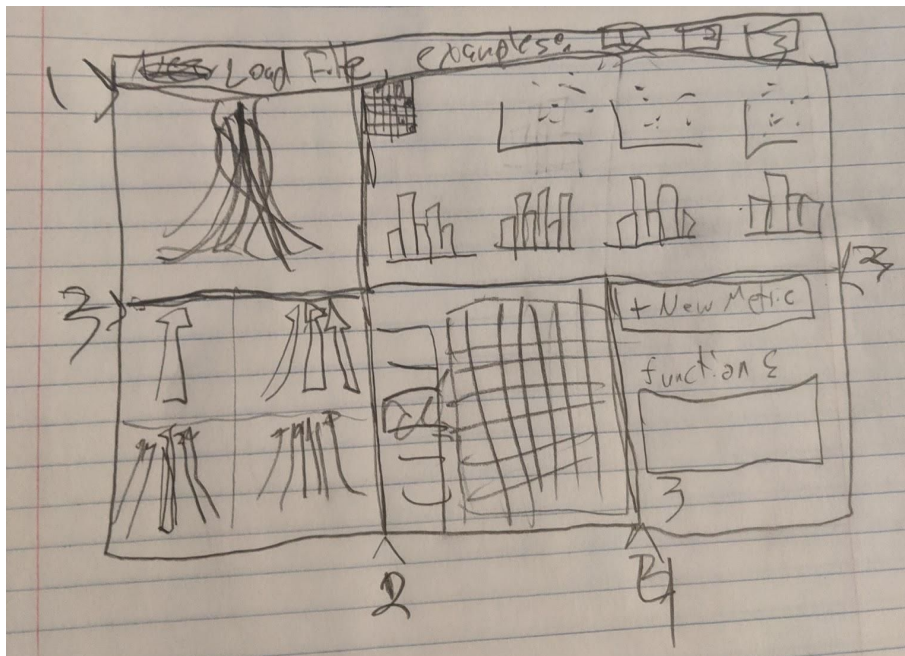
After writing the code for this layout framework as well as at least a stub class for each of the visualizations I can get a more accurately rendered picture of the final idea:
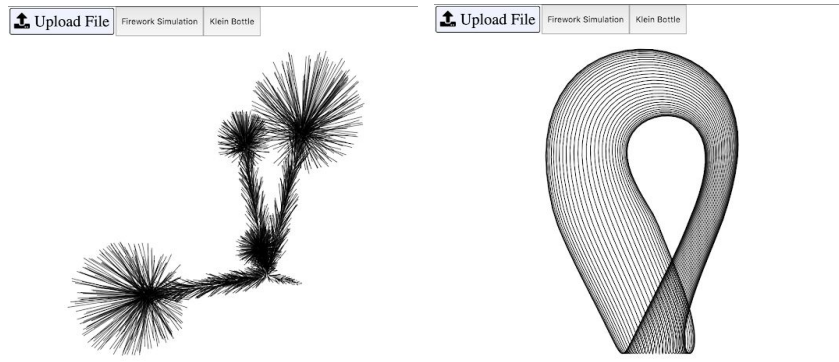
Two groups of people crossing at 90 degrees - with full screen for context:
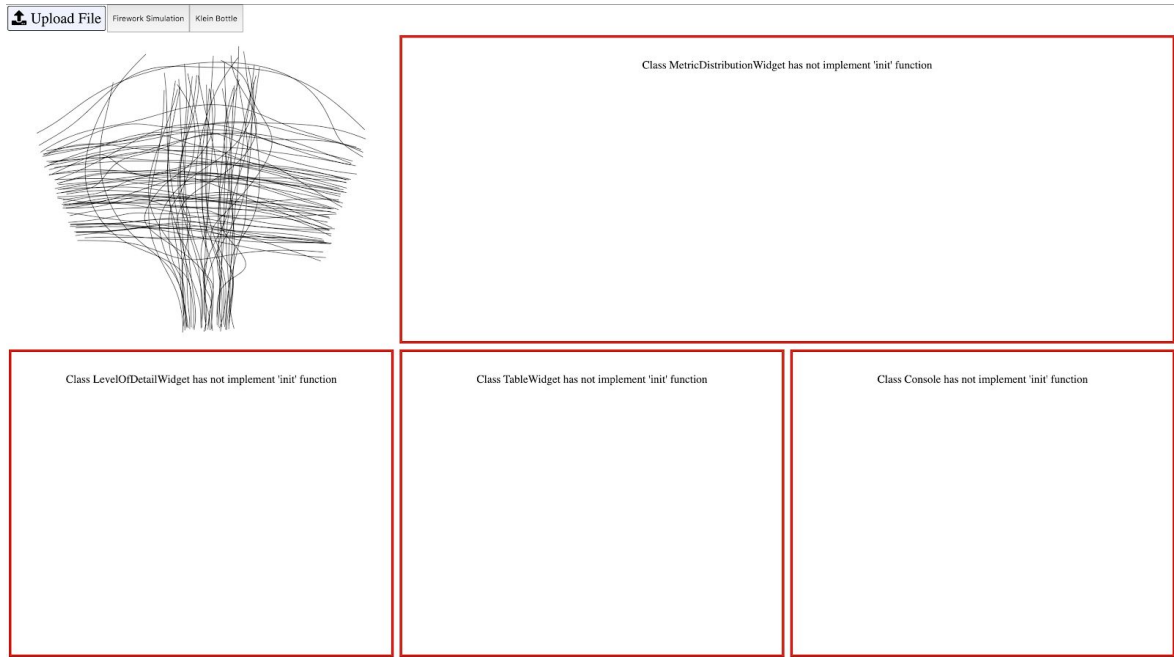
Other crowd scenarios:

Toy examples:

One insight I came on while working through scaling the data so that it keeps its aspect ratio intact is that instead of fitting data inside a box to it's fullest extent an argument can be made for shrinking a container to fit the data as tightly as possible.