Assignment 5

Continue working on your visualization tool. Note as a reminder that you are supposed to do more than what is asked in the assignments for your final project (see the project description in canvas). The assignments give you a guideline.

You should extend the interim report to the **final report**, for example, after this assignment you should be able to describe interaction techniques. In your final report, you need to mark the sections that you keep from the interim report such that your changes become clear. It should be obvious in the final report what has been changed.

Exercise 1 – Multivariate idiom

Extend your design and build more components according to the design. Implement further visualization techniques.

(a) Describe a task for which you need to see the relationship between multiple attributes (more than two/three) simultaneously, where you need the relationship with multiple attributes at the same time to develop the task. If you already have such a task then you already have it, just identify it, if not think about a task with these characteristics and add it to your set of tasks.

(b) Design and implement a visualization/idiom that would fulfill this task. The expectation is that you need a more complex idiom than the ones you have used until now.

(c) Justify the idiom, marks, and visual encoding choices in (b) given the task and the characteristics of the data attribute types involved.

Exercise 2 – Aggregation / Summarization per item

In the lectures, we have seen that if we have a large number of items to show, the direct visualization is cluttered and has issues due to occlusion. A way to avoid this issue is to make aggregations and summarize items.

(a) Select a task from the ones you have already defined or define a task that would benefit from applying an item aggregation strategy?

(b) Define, describe and justify a method to define groups/clusters of items, either by interaction selection/filtering or automatically (e.g., using k-means – it can be in a preprocessing step) that suits the task presented in (a). Notice that there are implementations available in python and JavaScript for common clustering techniques (e.g., k-means).

Clustering libraries

Javascript:
Kmeans: https://github.com/mljs/kmeans
DBscan: https://www.npmjs.com/package/density-clustering
Hierarchical: https://github.com/mljs/hclust

Python:
Scikit-Learn: https://scikit-learn.org/stable/
mljs/kmeans https://github.com/mljs/kmeans
K-Means clustering. Contribute to mljs/kmeans development by creating an account on GitHub. github.com

(c) Design a visual encoding to represent the groups or clusters of items by aggregation visualization (e.g., variations of boxplot). You do not want to show each individual item, but rather a summarization of the characteristics of the group or cluster. You can use standard methods but you can also generate your own new representation.

Exercise 3 – The Interaction Categories (in progress)

We have seen and analyzed some of the interaction techniques in our lectures. Interaction techniques integrated with your visualization tool can be used to show different views on the data or to refine parameters, for example, supporting you in building, refining, confirming, or rejecting hypotheses about the given dataset. In this exercise, we will take the 7 interaction categories introduced by Yi et al. as the basis and review your already implemented interaction strategies and potentially extend them.

Ji Soo Yi, Youn ah Kang, John T. Stasko, Julie A. Jacko: Toward a Deeper Understanding of the Role of Interaction in Information Visualization. IEEE Transactions on Visualization and Computer Graphics 13(6): 1224-1231 (2007)

Read the paper and briefly describe for each of the 7 interaction categories one example (just 2 sentences) of the interaction in your application. You can use this information to describe your interaction strategy in your final project solution. In the following, I will provide a list of possible interactions to give you some support

(I) **Select:** Let the user select bars or other elements in the visualization. If something is selected, highlight it, for example, with a complementary color

(II) **Explore:** Let the user scroll in one of the dimensions. For example, if you show a bar chart with values of an attribute visualization, let the user show different ranges by scrolling in one of the dimensions.

(III) **Reconfigure:** Let the user change the alignment, for example, if you used a staked bar you change the baseline to facilitate comparison.

(IV) **Encode:** Let the user change the representations, for example, the color coding, and shapes of the points or lines.

(V) **Abstract/Elaborate:** Let the user get details on demand, for example, when hovering over a point or bar, show a tooltip with extra details from the dataset.

(VI) **Filter:** Let the user filter for certain attribute intervals, for example, only over a certain threshold are displayed. You could use a range slider for that.

(VII) **Connect:** Let the user detect related elements. For example, by clicking on a point in a scatter plot highlight the same/similar values and the same/similar points in other views (brushing).

There are many options for building useful interactions, the above list is just a suggestion for you to give you some indication. Again, the visualization course is very open and you can be **creative** to build your own interactions, but be aware that you are expected to have at least one candidate for each of the 7 categories.