Introduction to Visualization
Fall 2018
Final Project
Process Book

Cole Sawyer and Michael Sawchuk
Professor Matthew Berger

Overview and motivation:

Our project aims to visualize uncertainty of datasets using bubble treemaps. This technique was outlined by Jochen Gortler, Christoph Schulz, Daniel Weiskopf, and Oliver Deussen in "Bubble Treemaps for Uncertainty Visualization" (http://graphics.uni-konstanz.de/publikationen/Goertler2018BubbleTreemapsUncertainty/bubble-treemaps.pdf). Our original motivation lied in the financial dataset that the paper implemented. As students interested in finance and data visualization, we sought to learn more about visualization techniques as well as learn more about stability of security prices. We were further motivated by an interest in general hierarchical data and novel ways to visualize this data. Through completing this project, we also learned much about using physics to create layouts as well as using Javascript libraries other than d3 such as Victor and planck-js. While the paper outlined many different ways to visualize uncertainty with bubble treemaps, we chose contour size as our main method. The larger the gap between the envelope contour and the circles, the more uncertain the data.

Related Work:

In class, we covered rectangular and squarified treemaps and briefly covered circular treemaps. The use of negative space to better show hierarchies in circular treemaps intrigued us, so we decided to use negative space to not only better see the hierarchy, but also to visualize uncertainty. In intermediate software design (CS 3251), we created a vector based physics engine, so we decided to continue explore its application through Victor and planck-js. We tried to find some good examples of using d3 for bubble treemaps online, but it proved to be very difficult to find some that were relevant to our specific technique.

Questions:

With our technique, we are trying to answer the question of how different data points in a hierarchical data structure compare while also considering the uncertainty of that data and being able to see the hierarchy they exist in. While these questions are not very difficult to answer individually, answering them all simultaneously requires a more complex visualization technique like the one we are using. These are the same basic questions we have been focusing on over the whole course of this project, but we have been more focused on different ones at different times. At the beginning, when we were still getting the basics of our vis completed, we focused mainly on visualizing the hierarchy by positioning the circles with the other ones in their groups, then we added the values as a visual channel with the radius of the circles, and then the uncertainty in the form of the contour. From there we had to further refine the hierarchy vis by making the program capable of dealing with additional layers of hierarchy.

Data:

For this project, we considered two datasets: price data on securities and starting salaries based on college majors. For our security price data, we used Yahoo Finance's datasets for stocks and ETFs, and used the St Louis Fed's online datasets for commodities. Since our data for one visualization came from two sources, it took a good deal of scripting to combine them.

While our data from Yahoo Finance provided daily open, close, high, and low prices, our commodity data was much less frequent and only listed closing prices. However, since we only cared about trends over several years, the frequency did not concern us so much. For the college major dataset, we were able to get all of the data from 538's online data source, but it still took a decent amount of work to get it into the same format as our financial data. This did not prove to be too difficult, as the data was given in such a way that it was easy to work with.

Implementation:

Our implementation of this visualization consists of several groups of circles, each surrounded by a contour, and then the contoured group forms another group with the other contoured groups, and so on for each level of hierarchy, though we only had two levels in our datasets to make it easier to work with and faster to compute. These groups of circles are each brought together by the d3 forcesimulation feature, and then the contoured groups are brought together using planck's joint functionality. The clusters of circles that form the contoured groups are treated as individual objects by planck with the way we have it set up. This results in the the visualization elements being centered in the screen and relatively space efficient. The contours themselves are designed such that each circle has a hidden larger circle associated with it, and the size of that hidden circle depends on the radius of the original circle and the enlargement which comes from standard deviation. The contour is formed by tracing the outer edges of the cluster of hidden circles such that it forms a single path.

When a user views our visualization, they can clearly see the different groups by seeing which circles are together in the same contour, and then they can hover over a circle to get more information about it, such as the actual values for the underlying data, in a details on demand approach. They will also see the name of the category which that circle is a part of. The intent of this is to allow users to choose which data points they care most about rather than cluttering the screen with all of the details at once, and they will still be able to make comparisons between that element and the data as a whole by looking at the visual channels we have used such as the enlargement that determines the spacing of the contours.

Difficulties with implementation:

Although there were several parts of this technique that proved to be very difficult, the most troublesome part was trying to take our groups of circles and position them relative to each other. Moving the individual circles was made much easier by d3's forcesimulation, but that did not work for our groups of circles, which meant we had to find another method. This is what caused us to look to the planck library as an alternate way to simulate forces, but it was quite difficult to learn how to use that library effectively.

Before we started using the d3 forcesimulation, we attempted to make our own code to simulate forces for the circles, as it seemed doable and we did not yet know about forcesimulation, but that proved to be incredibly difficult to make work perfectly, and we searched for a library to help us after much trial and error. Creating the envelopes around the circles was also very difficult, as the algorithm to do so is more complex than what we are used to programming, but fortunately the paper gave us something to start with for that.

Evaluation:

We evaluated our technique with a relatively similar method to that used in the paper, looking at the area perception, hierarchy perception, and runtime, but a key difference for ours is that we encoded the uncertainty with with area of the contours rather than the waviness of the contour, which required a different analysis, since we could not just count on the paper to have already given us a similar analysis. We observed the uncertainty encoded in our visualizations and attempted to use it to make sense of the uncertainty in the data, and we were able to establish general trends with good accuracy, though we still needed to rely on the details on demand approach to get a decent sense of what the specific values are.

Analysis:

Through applying this technique, we learned that bubble maps can be a visually pleasing and space-efficient way to represent data, but they can also make drawing conclusions about it slightly more difficult. For example, if we were to graph the prices of our different stocks, etfs, and commodities on a bar graph, we would very easily see which ones are highest and lowest and by how much, but when we put that data into the radius of a bunch of different circles that are positioned all over the screen it becomes slightly less automatic. However, that does not mean that this technique is bad, since in return for moving to a less effective visual channel for the values of the underlying data, we gain the ability to encode more attributes of the data in our visualization. For example, showing hierarchy in a bar graph is not as effective as doing so in our bubble treemap, especially when we have many levels of hierarchy.

Timeline:

Status at Update 1: We successfully found the financial data that we ended up using and formatted it into something we could easily process in our algorithm. We did have to make some changes to this later, but we kept a fairly similar structure to our data throughout the project. We did not manage to produce any visuals at this point.

Status at Prototype: Our prototype proved to be very incomplete compared to what we wanted to see. We were experimenting with the d3.pack algorithm at this time which we ultimately ended up not using, as it was too limited for this technique. The pack did give us something we could actually look at, but little of the code we produced at this stage actually ended up in the final product due to how many changes we made.

Status at Update 2: At this point, we were trying to work on a way to use our own force algorithms to position the circles and groups of circles, but we could not get the collisions to work properly, and circles either ended up overlapping or reacting violently to each other. We also did not have the contour algorithm working correctly, though we had started to work on it.

Status at Presentation: When we presented, we finally had the contour algorithm working correctly, which was important, as it is the key part of our technique. This meant that we had several separate groups with their individual circles grouped together with a contour around them, but the clusters were not all grouped together, meaning that we only had this working for the lowest level of the hierarchy.

Final Work: Between the presentation and the turning in of this project, we had a lot of improvements to make to the code. We changed several aspects of our force system and added new features in order to allow it to group the clusters correctly, and we made the code more generic so that we could tailor it to use a different dataset. Both of these were very non-trivial, but ultimately we proved successful, and an example of our vis for the college majors dataset is shown below. We also added a mouseover function so that when a circle is moused over, information about the corresponding node will be displayed in the upper right corner. When the user removes the mouse from the circle, the text disappears.