Assignment 2 report

Daniele Ferrario[1]

[1]University of Luxembourg

1 Introduction

This reports describe the implementation of two synchronized visualizations to represent multivariate data from a bike sharing dataset: one scatterplot and one hexbin scatterplot, both with 2D-Brush interaction to select multiple data objects.

2 Input data

The provided dataset contains 8760 records of bike-sharing related data from Seoul.

2.1 Quantitative attributes

1. Date
2. RentedBikeCount
3. Hour
4. Temperature
5. Humidity
6. WindSpeed
7. Visibility
8. DewPointTemperature
9. SolarRadiation
10. Rainfall
11. Snowfall

2.2 Categorical attributes

1. Seasons $\in$ {"Winter", "Spring", "Summer", "Autumn"}
2. Holiday $\in$ {"Holiday", "No Holiday"}
3. FunctioningDay $\in$ {"Yes", "No"}

3 Design choices

3.1 Scatterplot

As requested, the dataset is visualized through a scatterplot, which represent values for two quantitative dimensions, choosen through the form in the component *ControlBar*. All three categorical attributes are displayed through different visual variables.

Since the specifications didn't mention any order of importance of these attributes, I assigned the best visualization encodings with respect to the "Accuracy Ranking of Visual Variables" model, following the most natural way that I could think of. For the four seasons, I thought that color mapping would be the most suited visual representation, also because this is the most diversified attribute and it would result in a nicer visualization overall. For the remainings, *FunctioningDay* and *Holiday*, I've choosen respectively the shape of the item and the color of the border. (8465 "Yes" vs 295 "No", 433 "Holiday" vs 8328 "No Holiday"). Of course, other visualization mappings would still be valid choices.

3.2 Hexbin

As a second visualization I implemented a Hexagonal binning scatterplot. This type of representation is a 2D density chart that allows to visualize the combined distribution of two quantitative variables, which in this implementation can also be different from the scatterplot. This is done through the library *d3-hexbin*, which provides a method *d3.hexbin*(*pointsCoordinates*) that, given an array of points coordinates, returns an array of bins objects containing the reference to their grouped points, and the coordinate of the bin center. Being a density visualization, I sacrificed the visualization of categorical informations for the sake of clarity, prioritizing both color and size of the hexagons to describe densities. The colormap I used is Viridis, which efficacy is well known(1).

3.3 Brushing implementation

Being the second one a density visualization, the simultaneous highlighting of the selected items is not trivial, and needs additional data structures to map the items indices to the rendered points and the bins. The logic behind the selection is the following:

- Brush in **scatterplot**: When the brushing end event is triggered, the indices of the points in the selected window extent are saved in the state. In **hexbin**, every bin which contains **at least one** of the selected items is highlighted.
- Brush in **hexbin**: When the brushing end event is triggered, the bins which centers are positioned in the selected window extent are selected. Then **every point of the selected bins** are highlighted in the scatterplot.

These mappings are possible through the use of *indexToPointMap*, *pointToIndicesMap*, *pointToBinMap*, *binToPointsMap* maps in *HexbinD3* component.

4 Possible improvements

Some of the features can be improved. The second plot suffers from overlapping between the bins and the axis tick labels. A color legend for density could be added. Speed of rendering is affected by the high number of points.

References

[1] Yang Liu & Jeffrey Heer (*Somewhere Over the Rainbow: An Empirical Assessment of Quantitative Colormaps*)