uni.lu

UNIVERSITY OF LUXEMBOURG

Data Visualization of Housing Dataset

Name: Anisa Meta

Student ID: 024614503B

Github username: ameta20

Date: 04 November 2025

Dataset: Housing.csv

Visualizations: Scatterplot and Parallel Coordinates Plot

- The dataset I worked with is Housing.csv, which contains information about houses, including quantitative attributes such as price, area, bedrooms, bathrooms, stories, and parking, and categorical attribute called furnishingstatus (furnished, semi-furnished, unfurnished).
- This visualization assignment helps us explore relationships between these housing attributes and identify how certain features affect the price or size of a property or maybe other attributes.
- The scatterplot allows us to compare two quantitative variables at once (for example, area vs. price), while the parallel coordinates plot provides an overview of how multiple quantitative attributes interact. Both views are linked through interactive brushing, so that selecting data in one view automatically highlights selected items in the other.

First Visualization: Scatterplot

- In the Scatterplot, each circle represents a house which is mapped in the graph by the x value - area and y value – price. If a data point is not selected, its default color is black.
- By using the brush feature, we can understand easier how much data points fit inside a certain range of area and price. It can also help in defining clusters within the whole data points.
- To distinct the selected items, its color is changed to purple, and the opacity is changed to 1.
- By using scatterplot, we can easily detect the outliers.

Second Visualization: Parallel Coordinates Plot

- The parallel coordinates plot visualizes all multivariate attributes of the housing dataset. Each attribute has its own vertical axis and each house is shown by a polyline that connects the values across all axes.
- All axes are positioned horizontally using the d3.scalePoint (this.x) and there is an even distribution across the plot's width.
- This design helps to visualize many variables simultaneously.
- All categorical attributes' axes like mainroad, basement, airconditioning are positioned at the end of the plot. To handle their non-numerical data types, each unique category in an attribute's domain is assigned a discrete position in the corresponding axis.
- Furnishingstatus (Categorical Mark Coloring) is used to highlight a key categorical variable and it is encoded by the stroke color for each polyline. I have also provided a legend for the colors. By the help of this legend, I noticed an extra color, which can be a

sign of inconsistent data (like missing value). For this type of data, I handled by not showing them.

- When a line is hovered on, it is highlighted all in red to make it easier to read its values for each attribute.
- Like scatterplot, by brushing, the lines are highlighted into purple. Brushing in parallel coordinates plot is enabled only for one axis, so it is more readable.
- One axis line is partially hidden, showing only tick values for better visibility.

Synchronization between visualizations

The synchronization is handled in the by the App.js by sharing the selectedItems state variable among the both components. Whenever the brush action is activated and there is a change in the selectedItems, both visualizations (ScatterplotContainer and ParallelCoordinatesContainer) are updated.

D3 methods apply the visual highlighting by color and opacity changes to respective visuals circles and polylines.

Pros and Cons of the Design Choices

Pros

- Effective Synchronization between 2 visuals: Using brush in the parallel coordinates plot can help to also view in detail the area and price in the scatterplot.
- Consistent Visual Highlighting: For both visuals, I have used opacity 1 and color purple to distinguish them from other data points and lines. Also purple can be easily distinguished from other colors used in the second graph.
- The visual components respond to a click action to finalize the brushing selection.
- Using one categorical feature to color the polylines.

Cons

- If there would be more attributes, it would be more difficult to draw all vertical axes as the visual would be crowded. There would be overlapping in the lines, more difficult for readability.
- The visuals could have a zoom or tooltip feature for better readability.
- X and y attributes are hardcoded.