Interactive Housing Data Visualization: Synchronized Scatterplot and Sankey Diagram

Alberto Finardi
alberto.finardi.001@student.uni.lu
University of Luxembourg
Luxembourg

Abstract

This report describes an interactive data visualization system for exploring housing market data through two synchronized views: a scatterplot displaying continuous attributes (area vs. price) and a Sankey diagram representing categorical dimensions and their relationships. The system implements bidirectional brushing and linking interactions, allowing users to select data points in either visualization and observe corresponding highlights in both views. A key design decision involves treating discrete quantitative attributes (bedrooms, bathrooms, stories, parking spaces) as categorical variables due to their small finite domains, enabling effective flow visualization in the Sankey diagram. The implementation follows React and D3.js design patterns with centralized state management for cross-visualization synchronization.

1 Introduction

Understanding housing market data requires exploring both continuous attributes (e.g., price, area) and categorical features (e.g., number of bedrooms, amenities). This assignment implements two synchronized interactive visualizations to support multivariate data exploration: a **scatterplot** for analyzing continuous relationships and a **Sankey diagram** for visualizing categorical distributions and flows between dimensions.

The visualizations are implemented using React for component management and D3.js for rendering, following modern separation of concerns patterns. User interactions in either visualization are immediately reflected in both views through a centralized state management system, enabling coordinated exploration of the housing dataset from multiple perspectives.

This report describes the visual encoding choices, justifies the treatment of discrete numeric attributes as categorical variables, explains the interaction mechanisms, and discusses the strengths and limitations of the design.

2 Dataset Characteristics

The Housing.csv dataset contains 545 housing records with 13 attributes:

- **Continuous**: Price and area (used in scatterplot)
- **Discrete quantitative**: Bedrooms (1-6), bathrooms (1-4), stories (1-4), parking (0-3)
- **Binary**: Main road, guest room, basement, hot water, air conditioning, preferred area
- **Nominal**: Furnishing status (3 levels)

The discrete quantitative and categorical attributes (10 total) are treated as categorical in the Sankey diagram due to their small finite domains (2-6 distinct values each).

2.1 Rationale for Treating Discrete Numerics as Categorical

This design choice is justified by: (1) **small cardinality** (4-6 values per attribute), (2) **semantic discontinuity** (3 vs. 4 bedrooms represents distinct market categories, not a continuous scale), and (3) **visual effectiveness** - Sankey diagrams excel at showing flows and distributions across categorical dimensions, revealing patterns like "most 4-bedroom houses have 2-3 bathrooms" that would be obscured in continuous representations.

2.2 Evolution from Parallel Coordinates to Sankey Diagram

The Sankey diagram emerged after an initial **parallel coordinates** implementation failed due to severe overplotting. With only 2-6 distinct values per attribute, most data records collapsed onto identical parallel paths, making it impossible to distinguish relative frequencies or identify common versus rare transitions. The Sankey diagram solved this by encoding frequency in node heights and link widths, allocating vertical space proportionally to category counts, and providing clear flow visualization through ribbon-style links.

3 Visual Encoding Design

3.1 Scatterplot

The scatterplot uses standard positional encoding (x: area, y: price) with circles as marks. Selected points are highlighted with red strokes and full opacity, while unselected points have 0.3 opacity to reveal density patterns. This view was already implemented and serves as the continuous attribute exploration component.

3.2 Sankey Diagram: Categorical Flow Visualization

Key Design Elements:

- **Nodes**: Rectangles with height proportional to category frequency, sorted top to bottom by count. Three

state coloring: unselected (light gray `#d3d3d3`), dataflow (dark gray `#808080`), selected (red `#ff6b6b`).
- **Links**: Ribbons with width proportional to flow count between categories. Color scheme uses shades of blue: default flows (light blue `#64b5f6`, 0.2 opacity), selected flows (dark blue `#1976d2`, 0.7 opacity), hover state (medium blue `#42a5f5`, 0.7 opacity), dimmed flows (very light blue `#90caf9`, 0.2 opacity).
- **Labels**: Show category name and counts (e.g., "3 (127)" for 127 houses with value 3).
- **Dimension controls**: Drag and drop to reorder, click to add/remove dimensions (minimum 2 enforced).

The area encoding (node height, link width) effectively communicates univariate distributions, bivariate relationships between adjacent dimensions, and multivariate patterns across the full flow sequence.

4 Interaction Mechanisms and Synchronization

The system implements bidirectional brushing and linking with centralized state management (`selectedItems` and `selectionSource`) in the App component.

Scatterplot Brushing: Users drag a rectangular region to select multiple points or click individual points. The D3 brush component detects points within selection bounds, then automatically clears the brush visual (100ms delay) while preserving selection state. The axis domains include 5% padding for easier brush interaction. This is the primary new interaction implemented for this assignment.

Sankey Selection: Users click nodes (toggles selection) or links (selects both endpoints). The filtering logic implements **OR within dimensions, AND across dimensions**:

- **Within same dimension (OR):** Selecting "3 bedrooms" and "4 bedrooms" shows houses with *either* 3 or 4 bedrooms. This allows comparing different categories within the same attribute.
- **Across different dimensions (AND):** Selecting "3 bedrooms" and "main road: yes" shows only houses with *both* 3 bedrooms and main road access. This enables filtering to specific combinations.

This logic aligns with intuitive data exploration; users typically want to compare alternatives within a dimension but apply constraints across dimensions.

Synchronization: When scatterplot is brushed, the Sankey highlights affected nodes/links and updates labels to show "matching/total" counts. When Sankey nodes are clicked, filtered houses are highlighted in the scatterplot with red strokes and full opacity. The `selectionSource` tracking prevents conflicting updates.

5 Design Evaluation

5.1 Strengths

Complementary perspectives: The scatterplot reveals continuous correlations while the Sankey shows categorical distributions and flows, providing comprehensive coverage of heterogeneous attributes.

Flexible exploration: Dynamic dimension management (add/remove/reorder) enables hypothesis testing. Bidirectional linking supports natural workflows: "Show categorical profiles of expensive houses" (scatterplot -> Sankey) or "Where do 4-bedroom furnished houses fall on price-area?" (Sankey -> scatterplot).

Clear feedback: Three state node coloring and dynamic label updates ("matching/total") provide immediate selection feedback across views.

Effective categorical treatment: Reveals market segmentation patterns (e.g., dominant 2-3 bedroom with 1-2 bathroom combinations) that continuous representations would obscure.

5.2 Limitations

Scalability: Visual clutter emerges with 7+ dimensions or high cardinality attributes. Small frequency categories create thin, hard to interact with nodes.

Lost ordinality: Treating bedrooms/bathrooms as categorical discards ordinal information - "4 bedrooms" doesn't visually appear as twice "2 bedrooms."

Synchronization constraints: Brushing the scatterplot clears manual Sankey selections, preventing hybrid manual+automatic filtering workflows.

No statistical summaries: Users must visually estimate distributions without access to quantitative statistics (mean, median, quartiles).

6 Implementation Decisions

6.1 Custom Sankey Implementation vs. d3-sankey Library

A significant implementation decision was to develop a custom Sankey diagram from scratch rather than using the official `d3-sankey` library from the D3 authors. While the official library provides robust, production-ready functionality, the custom implementation was chosen for several reasons:

Educational Value: Implementing the Sankey layout algorithm manually provided deeper understanding of:

- Node positioning and height calculations based on data flow
- Link path generation using SVG Bézier curves
- Flow conservation principles in diagram layout
- State management for interactive highlighting and filtering

Customization Flexibility: The custom implementation enabled specific features tailored to this project:

- Fine-grained control over node and link styling with CSS variables
- Custom filtering logic (OR within dimensions, AND across dimensions)
- Seamless integration with React component lifecycle
- Specialized interaction patterns for bidirectional brushing and linking
- Dynamic dimension management (add/remove/reorder)

This approach prioritized learning outcomes and design control over using a pre-built solution.

6.2 Row-Based Layout for Optimal Interactivity

The application uses a row-based flexbox layout where visualizations are stacked vertically, with the scatterplot in the first row and the Sankey diagram in a larger second row. This design decision prioritizes the Sankey diagram's interactivity:

Rationale: The Sankey diagram is designed to expand horizontally along the x-axis, as dimensions are arranged left to right and flows connect between adjacent columns. Providing full horizontal width allows users to add multiple dimensions without horizontal scrolling and enables clear visualization of flow paths across many categories.

Implementation: The layout allocates 40% vertical space to the scatterplot and 60% to the Sankey diagram (using `flex: 2` and `flex: 3` respectively). This replaced an earlier column-based layout (using `.col2` with 50% width floats) that constrained the Sankey's horizontal expansion.

Scatterplot flexibility: Unlike the Sankey, the scatterplot's effectiveness is not orientation-dependent; it maintains its analytical value whether displayed in a wide or tall aspect ratio, making it suitable for the smaller first row.

7 Acknowledgments

The user interface styling and color scheme design were developed with assistance from AI tools, which provided suggestions for CSS gradients, spacing, and responsive layout patterns. The core visualization logic, data processing, interaction mechanisms, and React/D3.js integration were implemented manually based on course materials and D3.js documentation.

8 Conclusion

This project implements a synchronized dual view system combining a scatterplot (continuous attributes) with a Sankey diagram (categorical flows). A critical design choice was treating discrete quantitative attributes (bedrooms, bathrooms, stories, parking) as categorical, justified by their small finite domains and semantic discontinuity. This enabled flow visualization revealing market segmentation patterns that continuous representations would obscure, as demonstrated by the failed parallel coordinates prototype where severe overplotting made patterns indistinguishable.

The bidirectional brushing and linking interaction, implemented through React's centralized state management, creates a seamless exploratory experience. Users can fluidly ask "What categories characterize expensive houses?" (scatterplot -> Sankey) or "Where do specific categorical combinations fall in price-area space?" (Sankey -> scatterplot).

The implementation follows React and D3.js design patterns with clear separation between component lifecycle management and visualization rendering.