Interactive Data Visualization for Bike-Sharing Data

Tom Walter

November 26, 2024

Introduction

The goal of this project was to create two interactive, synchronized visualizations for exploring the bike-sharing dataset. I implemented a scatterplot and a density plot using React and D3.js. The synchronized brushing interaction connects both visualizations, allowing users to highlight selected data points in one plot and see the corresponding points in the other.

The dataset includes numerical variables like *RentedBikeCount*, *Temperature*, and *Humidity*, making it perfect for this type of visualization. Users can select any of these as the X and Y axes to tailor their analysis. The scatterplot focuses on individual data points, while the density plot summarizes data distribution. Together, these views create a dynamic and insightful data exploration tool.

Why I Chose a Density Plot

The density plot was a natural choice for the second visualization because it complements the scatterplot perfectly. Here's why:

- **Big Picture View**: While the scatterplot is great for analyzing individual data points and spotting outliers, the density plot provides a clear overview of where most of the data is concentrated.

- **Aggregated Insights**: It makes clusters and patterns immediately obvious, which is harder to achieve with just a scatterplot.

- **Clean Visual Encoding**: Density plots use smooth gradients to show concentrations, which makes them visually appealing and easy to interpret.

By combining the two, we get detailed analysis and an overall understanding of trends.

Why I Used the Viridis Color Scheme

I used the `interpolateViridis` color scheme from D3.js for the density plot for these reasons:

- **Perceptual Uniformity**: The gradient is designed so that differences in color correspond to differences in density in a consistent way. This ensures accurate interpretation.
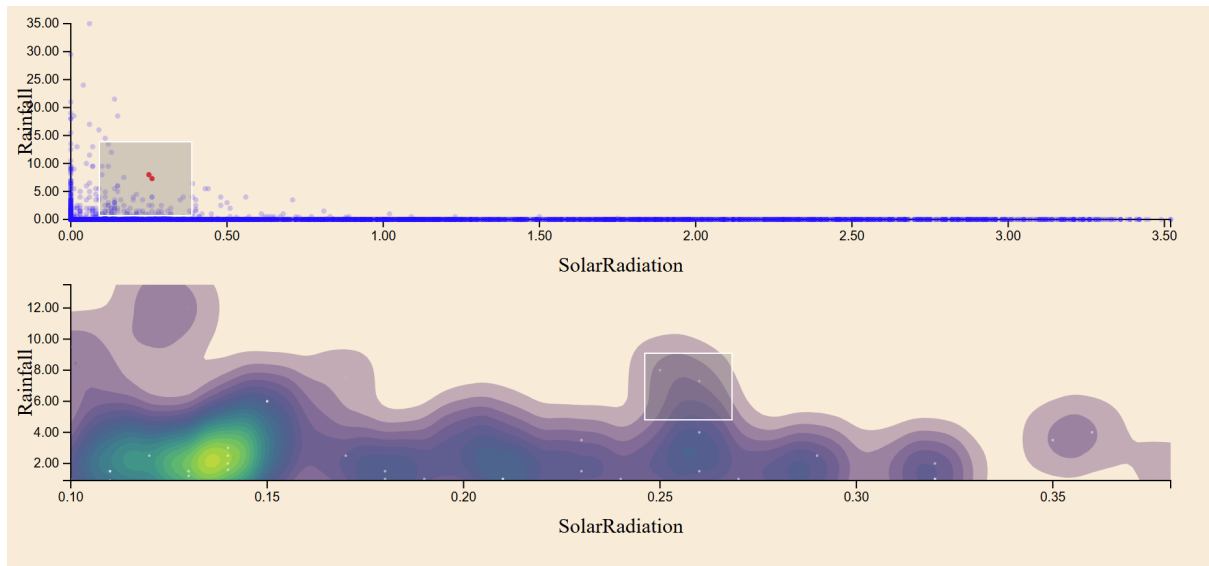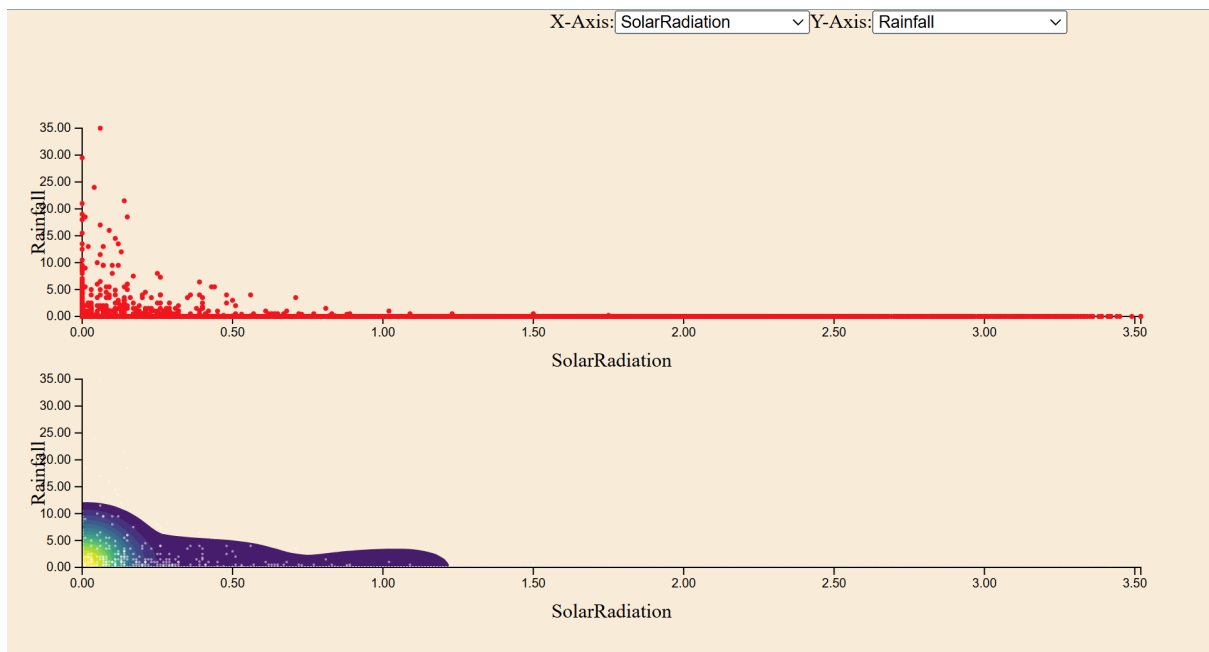
Figure 1: Simulataneous interaction with both 2D

Figure 2: Overview before Brush Use

- **Accessibility**: Viridis is colorblind-friendly, which is important for inclusivity.

- **Contrast**: The dark-to-light gradient stands out well against the scatterplot's blue and red points, making it easier to read both visualizations side by side.

Honestly, it also just looks good.

How the Interaction Works

Scatterplot Brush

In the scatterplot, users can select data points using a rectangular brush. Selected points turn **red**, standing out from the default **blue**. This makes it easy to see what's selected.

Density Plot Brush

The density plot allows users to zoom into a specific region using a brush. This updates the scatterplot, highlighting only the points in the brushed area. The two visualizations are fully synchronized, so any interaction in one is reflected in the other. This kind of dynamic linking helps users dive deeper into the data without losing context.

Pros and Cons of My Design

What Works Well

- You can interactively explore relationships between variables and compare patterns between the two visualizations.

- The scatterplot shows precise details, while the density plot summarizes the big picture. Together, they offer a complete perspective.

- Users can choose any numerical variables for the X and Y axes, making the tool flexible for different tasks.

- The red-highlighted points stand out clearly against the blue scatterplot, ensuring nothing gets lost in translation.

What Could Be Improved

- **Brush Synchronization Complexity**: Synchronizing brushes between two visualizations can occasionally cause slight mismatches. It's tricky to make it 100% seamless. I included the data points in the density plot with low opacity, but they are not rendered every time.

- **Limited to 2D relations**: Due to the nature of the density plot, it is only possible to visualize the relation between 2 dimensions of a data point at once.

Conclusion

This project showcases how scatterplots and density plots can work together to explore multivariate data interactively. The Viridis color scheme and synchronized brushing create an intuitive tool for analysis. While there's always room for improvement, I'm satisfied with the way the design makes it easy to see both detailed and big-picture patterns in the dataset.

You can find the full implementation on my GitHub: `https://github.com/tombart1997/Assignment2`.