Towards Fluent Interactive Data Visualization

Adam Bartonicek

2023-08-02

Introduction

There is a subtle yet profound issue in the production and use of interactive data visualizations: *when we interact with a plot, what exactly are we interacting with?* On the face of it, it may not even seem obvious that there is any problem. Like static plots, interactive plots consist of geometric objects such as points, lines, or areas. Surely, when interacting with a visualization, we can just interact with the geometric objects we see? However, this is rarely the case. The reason is that the geometric objects we see cannot be interpreted in isolation, but only with reference to some underlying data. Further, when plotting data, we rarely plot it directly. Instead, most plots use the geometric objects to display some statistical summaries of the data, such as count, sum, mean, or quantiles. These summaries are obtained by applying mathematical functions to the data. The core argument of the present text is that these mathematical functions (and their properties) impose limits on what kinds of visualizations and interactions can be meaningfully composed. Before diving deeper into the issue, however, let's first define some key terms and draw a rough sketch of the data visualization process.

To create a data visualization, be it static or interactive, we need several key ingredients: data, summaries, scales/coordinate systems, and geometric objects. Firstly, every data visualization is built on top of some underlying data. Typically, the data is stored in a tabular format which we can represent as a set of rows R (this is not strictly necessary, but will be used here for convenience). Secondly, each row $r_i \in R$ is transformed via a mathematical function into a collection of summaries $s_j \in S$. The function may be one-to-one (bijection), as in the case of a scatterplot, or, more often, many-to-one (surjection), as in the case of plots such as barplot, histogram, density plot, or violin plot. The function will typically reduce the cardinality of the set at hand such that $|S| \leq |R|$ (i.e. in a typical barplot, there will be fewer bars than there are rows of the data). This is done by stratifying on some variable which may either come from the data directly (as in the case of a barplot or a treemap) or may itself be a summary of some variable in the data (as in the case of histogram bins). Importantly also, each collection $s_j \in S$ may (and usually will) hold multiple values produced by a different constituent function each - for example, the collection s_j of summaries for a single boxplot box will consist of a median, first and third quartile, and the minimum and maximum of some variable, for a given level of some stratifying variable (which itself is an element of s_j). The output of these constituent functions may also depend on some extraneous parameters (such as anchor and binwidth in a histogram). Combined, the summaries form a partition of the data - if we give each collection of summaries a label $j \in J$ such that $|J| = |S|$, we could split the data rows $r_i \in R$ into sets belonging to the same collection of summaries R_j (with no overlapping parts: if $j \neq k, R_j \cap R_k = \varnothing$), such that, if we combine these sets, we recover the original dataset: $R = \bigcup_{j \in J} R_j$. Thirdly, each collection of summaries $s_j \in S$ needs to be translated from the data- (or summary-) coordinates to graphical coordinates/attributes $g_j \in G$, by transforming each value via a scale (note that this mapping preserves cardinality: $|G| = |S|$). For numeric summaries, these typically come in the form of a linear transformation, such that the minimum and maximum of the data are mapped near the minimum and maximum of the plotting region along some axis, respectively. The scales may also provide additional transformations such as log-transformation or binning. Finally, the collections of graphical coordinates $g_j \in G$ are displayed by plotting them as geometric objects. These may be simple, such as points, lines, or bars, or compound, such as a boxplot or pointrange.

The whole process can be summarized as such:

$$R \to S \to G$$

$$\text{(rows)} \to \text{(summaries)} \to \text{(graphical coordinates/attributes)}$$

The above should be fairly non-controversial description of how a data visualization is produced, and applies equally well to static as well as interactive visualizations.