

It Ain't Necessarily So: Checking Charts for Robustness

Aran Lunzer*

Viewpoints Research Institute

Amelia McNamara†

University of California, Los Angeles

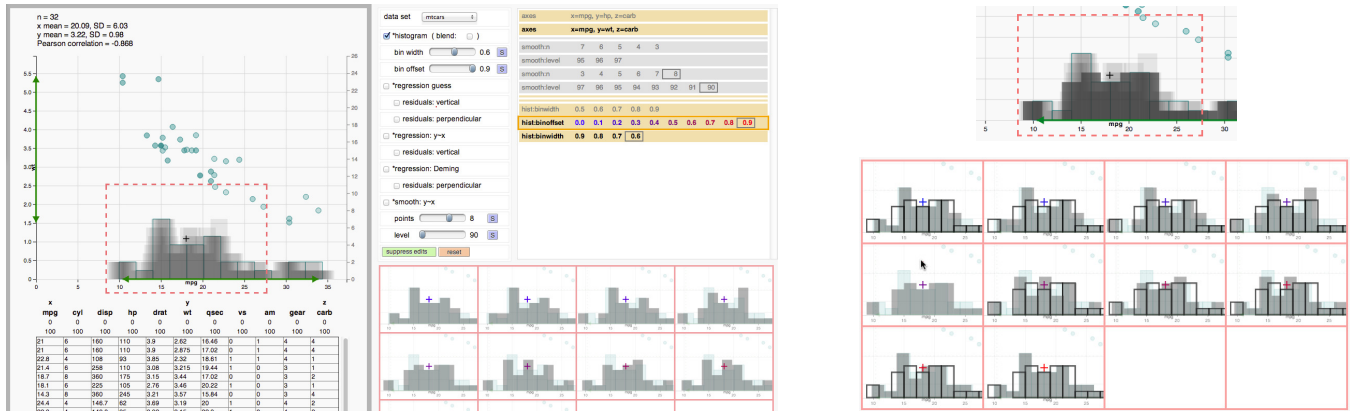


Figure 1: Checking how bin offset influences the shape of a histogram for a small data set. Histograms for ten offset values are combined on the main chart using translucent overlay; the user has also steered the dashed-red probe region to display this part of the respective charts in a small-multiples view. On the far right we show how mousing over one of the small-multiples panes highlights the corresponding histogram on the main chart, and also shows the outline of that histogram on the other panes.

ABSTRACT

Graphics such as statistical charts are crucial tools for communicating the essence of data, but require care in their construction: a chart for a given data set could appear to tell a range of stories, depending just on parameter values used in generating the chart. We are investigating how providing an interactive history of chart-parameter manipulations could encourage people to check the robustness of any story a chart appears to show. Our hope is that this work will be of value to chart creators and consumers alike.

Keywords: Small multiples, comparisons, interaction history, sub-junctive interfaces, multiple views, exploratory data analysis.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces (GUI);

1 INTRODUCTION

As experts accustomed to creating and reading data graphics, we’re all familiar with ways of lying with visualisation. And certainly, those who want to mislead with statistics have a battery of methods at their disposal [1]. However, there are many situations where a finished graphic may be misleading simply because the author did not have a full understanding of their data, or of parameters used in specifying the graphic.

As interactivity becomes more common across all media sources, there are increased opportunities to highlight the fragility of analysis both during and after the creation of a finished product. It also means that rather than making one static graphic or table, data journalists can (and do) publish large-scale interactive

tools that allow users to explore and discover other ‘stories’ in the data themselves. We are experimenting with features that could be incorporated into data visualisation tools to capitalise on this interactivity, aiming to help users explore the space of visualisations that could be derived from a particular data set. Our field of interest is statistical charts—where even simple abstractions such as histograms can be drawn misleadingly, and carelessly chosen measures such as linear regression are often fragile with respect to data sampling or due to rogue outliers.

The initial experiments reported here are focussed on support for rapid, incremental generation of alternative visualisations, and making comparisons between them. Our goal is to encourage users to explore the impact of the settings and parameters, helping to answer the question ‘How robust is this presentation of the data?’.

2 OUR EXPERIMENTAL PLATFORM

The screen shots in this abstract are taken from a charting experiment that we ran in LivelyR, a platform we built for exploring interactive statistical charts generated by R. LivelyR is built using the Lively Web development environment [2], and connects to an R server (typically running on the same computer) through a customised version of the R package ggvis¹. In this experimental setup the user specifies the chart seen on the left in Figure 1 using the custom set of sliders and checkboxes in the central panel of the interface. Dragging a parameter slider causes immediate refresh of the chart for each new value encountered.

At top-right in the interface is the history list, which is the key to our support for parameter exploration. The history list is an accumulated record of the user’s chart-parameter adjustments.

2.1 A rewindable, reusable history

The history list grows downwards, adding one item for each continuous manipulation of one parameter—such as the dragging of a

*e-mail: aran@acm.org

†e-mail: amelia.mcnamara@stat.ucla.edu

¹A demonstration of some of the early capabilities of LivelyR can be viewed at <http://bit.ly/livelyr-intro>.

slider over some portion of its range. Every value passed to R for reconfiguring the chart is recorded, though repetitions of a value are filtered out. When a parameter has been adjusted many times, it will be represented by multiple items in the list; the value currently being used in the chart is the rightmost entry in the last of those items. Earlier items for such a parameter are shown using grey text, while items for parameters that are no longer contributing to the chart (such as bin-width settings on a chart for which the histogram is currently switched off) are entirely greyed out.

Accumulating the history does not interfere with the manipulation of the chart, but makes possible various facilities for reusing the manipulations carried out so far.

Figure 2 shows History Rewind, which enables random-access revisiting of any point in the sequence of interactions. Pointing to a value within a history item causes the chart to revert to the state it had when that value was picked: all parameters other than the one whose item is being revisited take the values they had at that point, and all later manipulations are ‘in the future’ and ignored.



Figure 2: History Rewind. In the session shown in Figure 1, the user adjusted parameters for a smoothed line, then switched off the line and switched on a histogram. Here the ephemeral history rewind is being used to revisit the first time the histogram bin width was set to 0.6. History items in the future are shown in a lighter colour: for example, here the bin-offset slider has yet to be moved from its default initial value of 0.

A second facility is History Override. Instead of rewinding history to an earlier state, this is an ephemeral modification of the final state of the chart—the state corresponding to the current ‘end of history’. In this case, pointing to a history-item value overrides the final value the parameter would otherwise have. In our running example, the chart’s final state has histogram bin width of 0.6 and offset 0.9; a history override on a bin-offset entry with value 0.2, say, will temporarily apply that value to the chart. Note that this is not a state that has been seen before, because when the user was first manipulating the bin offset the width was 0.9. It’s therefore a quick way of testing a counterfactual: ‘How would the chart look now if I had released the bin-offset slider on 0.2 instead?’.

2.2 Parallel exploration using parameter sweeps

Rather than having to explore a parameter’s effect one value at a time, we allow the user to set up a ‘sweep’ that results in a combined display of the outcomes from a range of values. Figure 1 shows how a sweep on the bin-offset parameter causes a family of translucent-grey histograms—one for each value of the parameter—to be generated and overlaid on the chart. The resulting display is fuzzy, showing where the histograms do and don’t overlap. This alerts the user to the impact of this parameter, and hence the dangers in drawing conclusions from any single histogram.

Similarly, in Figure 3 the user has set up a sweep on a range of values for the number of prediction points used to generate a smoothed line through a scatter plot. Again, the extent to which the lines and their confidence regions fail to overlap shows the influence of this parameter.

In accordance with the principles of subjunctive interfaces [3], any adjustment to a parameter other than the one with the sweep will apply to all the sweep values in parallel. For example, in Figure 1 the user could perform a history override on bin width to see whether the overlaid histograms for a width of 0.7, say, are as fuzzy as the histograms for 0.6. This greatly accelerates the process of exploring diverse parameter-value combinations.

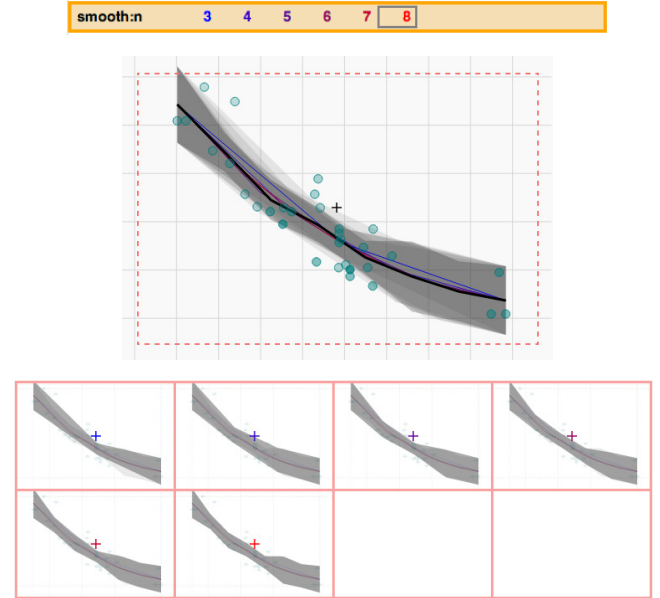


Figure 3: Parameter sweep on the number of prediction points n for a 2D model-fitting line. A sweep of six values of n has been set up, and the resulting lines and their respective confidence intervals overlaid; the current setting ($n = 8$) is drawn using a wider, black line. A Region Probe shows their individual shapes side by side for more detailed inspection of the differences.

2.3 Fine-grained comparison using small multiples

A fuzzy set of overlaid results reveals that the swept parameter has a major impact, but it can be hard to grasp the details of how the results differ. In addition to the overlay we therefore offer a small-multiples view (shown in both Figures 1 and 3), each pane of which shows one version of the chart in the region of the movable dashed-red box. Figure 1 shows how mousing over an individual pane causes that pane’s chart version to be highlighted in the chart view, and also to appear in outline on the other panes to further help the user make comparisons.

We believe that supporting both overlaid and side-by-side presentations of alternative charts will make these facilities useful for a wide range of chart types. Our experiments continue.

ACKNOWLEDGEMENTS

We are happy to acknowledge collegial and financial support from the SAP Communications Design Group and from Viewpoints Research Institute. We particularly thank Robert Krahn for helping us to build with Lively Web.

REFERENCES

- [1] D. Huff. *How to lie with statistics*. W W Norton & Company, 1954.
- [2] D. Ingalls and R. Krahn, et al. The Lively Web. <http://lively-web.org/>, 2013.
- [3] A. Lunzer and K. Hornbæk. Subjunctive interfaces: Extending applications to support parallel setup, viewing and control of alternative scenarios. *ACM Trans. Comput.-Hum. Interact.*, 14(4):1–44, 2008.