Looks Good To Me: Visualizations As Sanity Checks

Michael Correll, Mingwei Li, Gordon Kindlmann, and Carlos Scheidegger

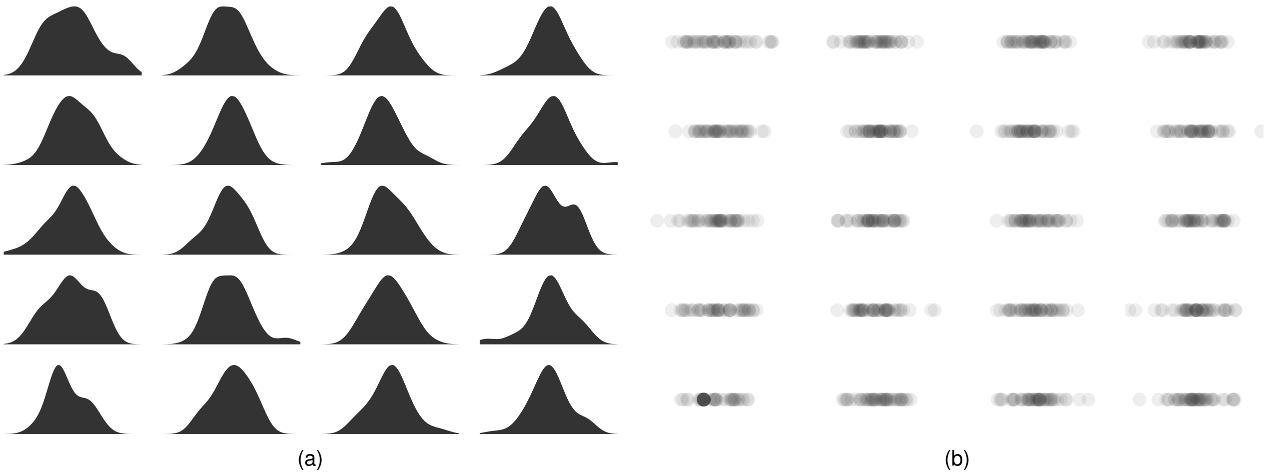


Fig. 1: Example lineups from our evaluation. Both Fig. 1a and 1b show the same univariate datasets. 19 of these charts are "innocent" random samples from a Gaussian. One "guilty" chart is mostly random draws, but 20% of samples have an identical value (an extraneous mode). The oversmoothed density plot makes this abnormality difficult to see (participants were only 35% accurate at picking out the correct density plot). Low opacity dot plots, however, make the dark black dot of the mode easier to detect (85% accuracy). See §5.1 for the right answer.

Abstract—Famous examples such as Anscombe's Quartet highlight that one of the core benefits of visualizations is allowing people to discover visual patterns that might otherwise be hidden by summary statistics. This visual inspection is particularly important in exploratory data analysis, where analysts can use visualizations such as histograms and dot plots to identify data quality issues. Yet, these visualizations are driven by parameters such as histogram bin size or mark opacity that have a great deal of impact on the final visual appearance of the chart, but are rarely optimized to make important features visible. In this paper, we show that data flaws have varying impact on the visual features of visualizations, and that the adversarial or merely uncritical setting of design parameters of visualizations can obscure the visual signatures of these flaws. Drawing on the framework of *Algebraic Visualization Design*, we present the results of a crowdsourced study showing that common visualization types can appear to reasonably summarize distributional data while hiding large and important flaws such as missing data and extraneous modes. We make use of these results to propose additional best practices for visualizations of distributions for data quality tasks.

Index Terms—Graphical perception, data quality, univariate visualizations

1 Introduction

The visualization of distributions of variables along particular data dimensions is a critical step in Exploratory Data Analysis (EDA) [47]. Summary statistics, by themselves, may not capture important aspects of the data [6, 34]. For large and complex datasets, visualizations of distributions work as "sanity checks" by providing evidence that the underlying data are reasonably free of flaws such as missing values or excessive noise that might affect later analysis, and by informing hypotheses about interactions and relationships among variables. Sanity checks also build trust in the underlying data processes. These sanity checks assume that data flaws produce characteristic and legible visual signatures in visualizations. This paper investigates the validity of that assumption by studying how well different distribution visualizations show data flaws, and by showing that choices (possibly careless or malicious) of visualization design parameters can make the visual signatures of data flaws more or less prominent.

We frame our study in terms of Kindlmann and Scheidegger's Algebraic Visualization Design (AVD) [30]. AVD considers visual encodings in terms of *possible alternative worlds*: if the data were different in an important way, how would that affect the visualization? A change in the dataset, depicted as a transformation $\alpha : D \to D$ in data space, is paired via the visual encoding with precisely one transformation in image space $\omega : V \to V$. In this paper, we define the data change α as adding some flaw to the otherwise "clean" data. We then assess the legibility of the corresponding visualization change ω by using Wickham et al.'s line-up protocol [49] to measure the accuracy with which viewers can detect flawed data by identifying its visualization among visualizations of clean data instances.

Concretely, we show through simulations and a crowdsourced study that different data quality issues (such as extraneous modes, missing values, and outliers) have differing levels of detectability across standard visualization techniques (such as histograms, density plots, and dot plots). Furthermore, the detectability depends strongly on visualization design parameters (such as the number of histogram bins,

- *Michael Correll (mcorrell@tableau.com) is with Tableau Research.*
- *Mingwei Li (mwli@email.arizona.edu) and Carlos Scheidegger (csheid@csheid.net) are with the University of Arizona.*
- *Gordon Kindlmann (glk@uchicago.edu) is with the University of Chicago.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

kernel bandwidth, and mark opacity): many visualizations that look "reasonable" can effectively hide data quality issues from the viewer.

This paper therefore functions as a sort of vulnerability analysis: we show it is possible to create visualizations that seem "plausible" (in that their design parameters are within normal bounds, and they pass the visual sanity check of revealing nothing untoward in the underlying data) but hide crucial data flaws. We show that no one standard visualization design is robust to all of these sorts of attacks, but that each have different patterns of vulnerability. We conclude with recommendations that will improve the robustness and reliability of visual sanity checks.

2 Visualizations of Distributions

In this work, we focus on visualizations of univariate distributions for various reasons: EDA often starts with creating univariate distribution visualizations, they are simpler (by having fewer design parameters) than multivariate ones, and they have been extensively studied in both the statistics and visualization communities. Therefore, while we acknowledge that sanity checks can occur at all scales and steps of the analysis process, we use univariate visualizations as a testbed for illustrating that data flaws can have a complex and potentially problematic relationship with visual features. A full review of all techniques for visualizing distributions is outside the scope of this paper, so we discuss a small set of visualization types commonly encountered in EDA tools, focusing on the design parameters that impact their visual design, and how these parameters are set by default.

The visualization deficiencies we highlight may be ameliorated with interactive user-driven adjustment of design parameters, but we argue that this does not solve the fundamental issues for visualizations as sanity checks. Interactively setting parameters for sanity checking requires the analyst to manually explore the design space for each data dimension independently, which is expensive in terms of time and the analyst's attention. On the other hand, default parameter settings (data-driven or otherwise) can produce "plausible" visualizations that nevertheless hide important issues. The analyst may not know that there are features that could be revealed through interaction, and would therefore have no incentive to alter the visualization.

We maintain that the default settings of designs parameters have an outsized influence on how data dimensions are spot-checked, so we consider default parameters across several common types of univariate visualizations in common visual analytics systems.

2.1 Histograms

Data preparation and summarization often uses histograms, which count data occurrences in discrete bins (as in Fig. 7). The main design parameter is therefore the size and location of bins. Using too many bins can produce sparse or noisy counts that obscure the overall distribution shape, but with too few bins, spatial modes or trends may be lost.

Seemingly reasonable default settings for histogram bin size can be computed by various "rules of thumb", based on statistical assumptions that may or may not hold for real datasets. For instance, Sturges' rule for histogram bins is derived from an assumption that the distribution to be estimated is a unimodal Gaussian [42]. This results in over-smoothed histograms when the data are not normal (for instance, if the data are bimodal, or have long-tailed outliers).

Data analysis systems may also *support* more complex methods, but simplistic rules tend to be the default. For example, Sturges' rule is the default for R [3] and D3 [1], and a modified form of the Freedman-Diaconis rule [18] is the default histogram binning method in Tableau [4]. Lunzner & McNamara [32] explored "stretchy" histograms that allow the user to interactively compare histogram binning parameters, but acknowledge that very few standard data analysis tools fluidly support interactive re-binning with immediate visual feedback.

2.2 Density Plots

Density plots use an underlying Kernel Density Estimate (KDE) to create a smooth curve based on the observed data (as in Fig. 1a). The type and bandwidth of these kernels has a large impact on the resulting plot. As with histograms, there is a tradeoff between choosing a bandwidth that is too large (and so over-smooths the KDE) or too small (and so under-smooths the KDE) [41].

The default kernel bandwidth setting is also frequently based on rules of thumb, such as the commonly used Silverman's rule [43], the default in R [2]. More sophisticated data-driven methods for selecting the bandwidth exist (such as Sheather & Jones's pilot derivative-based method [44]), but these often require complex calculations that may not scale to large numbers of points, or they require sampling or other approximation techniques that may not be stable across runs (see Jones et al. [26] or Park & Marron [37] for a survey). Silverman's rule, as with Sturges' rule, assumes that the underlying sampling distribution is a unimodal Gaussian, and so can oversmooth data where these assumptions are violated. With the exception of 2D density heat maps (which typically have adjustable bandwidths), we are not aware of any common EDA system that affords the interactive setting of kernel bandwidths with immediate visual feedback.

2.3 Dot Plots

Dot plots and strip plots encode each sample of a distribution as a dot or a line, respectively (as in Fig. 1b). They can be considered as a special one-dimensional case of scatterplots. While we focus on dot plots in this paper, our guidelines apply to most visualizations of distributions where one sample directly corresponds with one mark.

Overplotting in scatterplots can obscure distributional data. This is especially true for dot plots: due to the curse of dimensionality, overplotting is more likely to occur in univariate samples of a distribution than in bivariate samples of the same distribution. One technique to reduce overplotting is to jitter the marks, as in a beeswarm plot [17]). Adding jitter increases the vertical space taken up by the plot, and might be impractical if there are a large number of points to be plotted.

Altering the mark opacity is a common way to reduce the effects of overplotting. However, just as with histogram binning and KDE, this opacity must be chosen with respect to the structure of the data. With too much opacity, the modes and the shape of the distribution become invisible with overplotting. With too little opacity, outliers and smaller structures are impossible to see. Recent approaches to optimize mark opacity in scatterplots rely on screen space image quality metrics [33, 35] and have not seen wide adoption in common visualization systems. The default opacity of marks in EDA systems we examined is either fully opaque (as in R and Tableau), or a constant opacity (for example in Vega-Lite [40], the default opacity of non-aggregated marks is 0.7 [5]). Unlike our previous visualization examples, common EDA systems like Tableau, Excel, and PowerBI *do* allow the interactive manipulation of dot plot opacity through sliders with immediate visual feedback. However, in contrast, to our knowledge, no common visual analytics system provides data-driven procedures for optimizing dot plot opacity, or provides data-driven defaults.

Reducing the spatial extent of marks can also address overplotting, either by reducing mark size, or by replacing closed shapes with open ones. In Tableau and Vega-Lite, for instance, the default dot plot mark is an open, rather than filled, circle. However, as with opacity, the default *size* of the mark in dot plots is usually a constant, rather than a data-driven, value. There is also an interplay between size and opacity: the perceptual discriminability of the lightness of marks (and so, implicitly, the density) becomes poorer as the marks become smaller [45].

2.4 Other Univariate Visualizations

Other strategies for visualizing distributions and sampled data avoid presenting the samples directly, but instead visualize various summary statistics, such as showing the mean with a dot and variance with error bars. As another example, boxplots can communicate quartile information, and can also be extended to communicate more complex summary statistics [39]. However, for the same reasons that summary statistics alone are not sufficient as sanity checks [6, 34], visualizations of these summary statistics by themselves may also fail to indicate important data properties or flaws. In addition, correctly interpreting summary statistics can require knowledge outside the analyst's expertise; error bars in particular are often misinterpreted [8, 11].

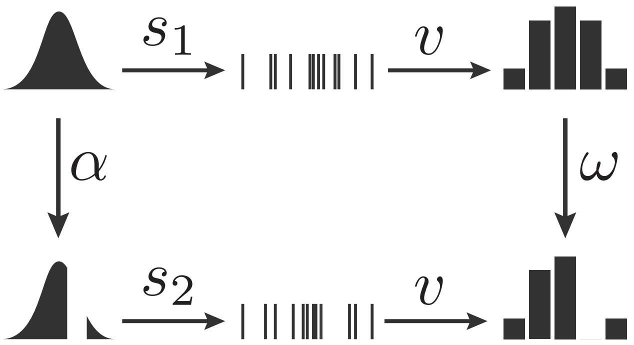


Fig. 2: The commutative diagram in this figure provides a way to evaluate visualizations of population samples. If a change α occurs in the population distribution (for instance, the introduction of a flaw such as a region of missing values), we expect to see a proportionally large change in the visualization ω. However, the data to be encoded is often a sample of this unknown distribution: this sampling s can also introduce visual changes in the resulting mapping from data to visualization v.

Of particular interest to us are "hybrid" or "ensemble" visualizations of distributions: these classes of visualization either combine layout and plotting techniques from multiple designs into one novel design, or juxtapose or superimpose multiple chart types together, respectively.

Wilkinson's dot plot [50] is a hybrid visualization that combines dot plots and histograms by placing marks into discrete columns, stacking marks when there would otherwise be overplotting. This method replaces the parameter of histogram bin width with that of the mark size: changes in mark size can cause large shifts in where columns are laid out. As with beeswarm plots [17], Wilkinson dot plots also take up more space, depending on the mark size and modes of the distribution.

As examples of ensemble charts, violin plots [22] combine a density plot with a box plot, and bean plots [27] combine a density plot with a strip plot. These combinations are meant to supplement the deficiencies of their component visualizations, but are somewhat ad hoc: it is not clear what sorts of ensembles best support different sorts of sanity checks. Designers must also independently set parameters for each of the component visualizations.

3 Algebraic Visualization

Kindlmann & Scheidegger's framework of algebraic visualization design (AVD) suggests two principles for creating and evaluating visualizations [30]. First, visualizations should be assessed by reference to a potential *transformation of the data*, denoted in AVD by α. Every function α that transforms the input induces a function that transforms the actual image produced by the visualization, denoted by ω. By considering different ways to transform the input, designers can investigate how the visualization responds. In AVD, a change in data α that does not change the visualization output (that is, whose corresponding ω is the identity) is termed a *confuser*. Confusers identify ambiguities in visual mappings by capturing differences in the data that are invisible to the viewer. Crucially for AVD, αs that are confusers for some visual mappings are *not* confusers for other visual mappings. Second, AVD asserts that visualizations should be *invariant to equivalent representations of the same data*. If a visualization produces different outputs because of the order in which elements (representing a set) are stored in a list, then the switch from one representation to another is a *hallucinator*: a way to "trick" the visualization into producing an apparent change where there should be none.

Our work builds on the observation in AVD that the *sampling* used to obtain finite samples from an underlying population can be thought of as a representation *mapping*. Figure 2 indicates samplings s_1 and s_2 from different data distributions; there are clearly many other possible sampling mappings from each distribution. If a nontrivial data change α commutes with an identity ω (i.e. no visual change), then visualization v is discarding information from the data that is necessary to disambiguate the effect of α. On the other hand, if α is the identity while ω is not, then the visualization is showing superficial differences between the different samplings of the same data population, rather than a feature of the data itself.

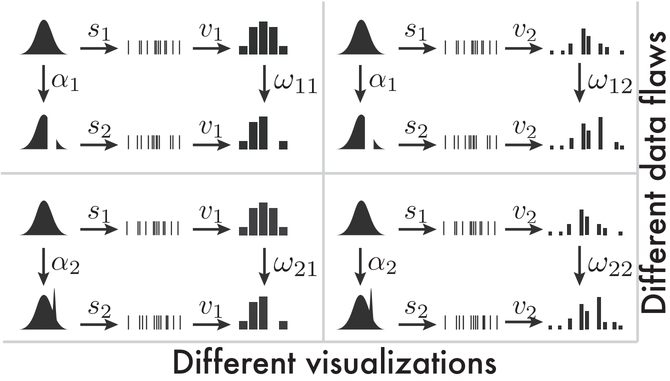


Fig. 3: Different data flaws interact with different visualization mappings differently based on the setting of design parameters, producing a large number of potential ωs. A visualization and parameter setting that is adept at depicting a certain class of data flaws unambiguously might be a confuser for another type of flaw.

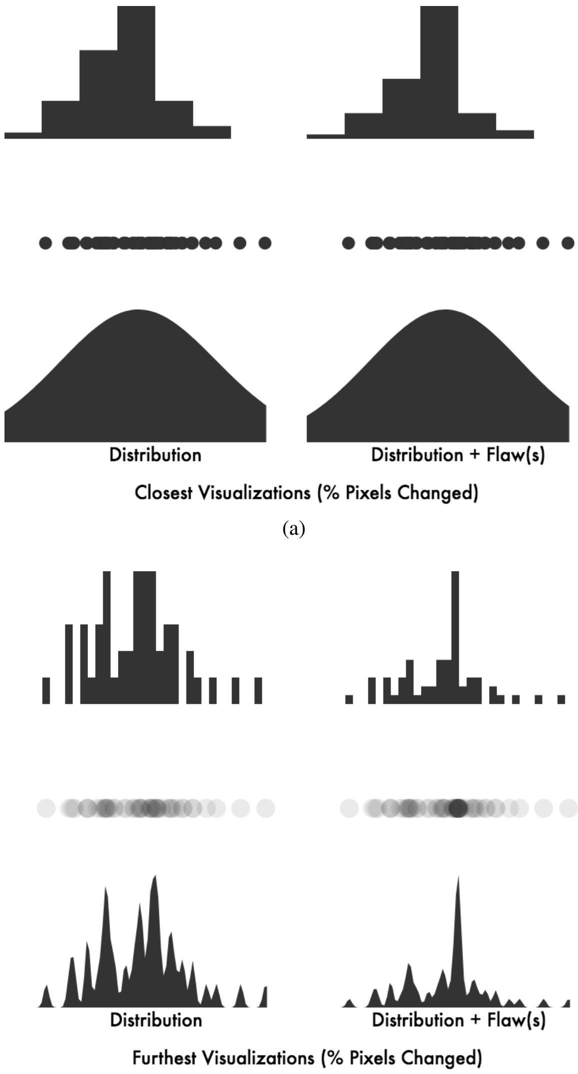
Our study models data quality issues (missing data, repeating values, more outliers) as different αs, and then creates many visualizations without those flaws but with different sample mappings. If participants are unable to spot the odd visualization, then one of the two failure modes described above has occurred: either a significant alteration in the data failed to produce a large enough change in the visualization to make it stand out (a confuser), or the visual differences caused by non-significant sampling variation was sufficient to produce visualizations that stand out despite having no data alterations (a hallucinator). Note that, as we will describe in § 5.1.4, our current study cannot distinguish between these two failure modes.

In addition, our work highlights a complication in applying AVD to the design and evaluation of real-world visualization designs. AVD seeks to reduce the number of arbitrary decisions in the course of visualization design (or, conversely, to point out arbitrary decisions in potentially sub-optimal designs). The main discussion in AVD focuses on forced choices and arbitrary features of input representations being the cause of confusers and hallucinators. Notably, AVD does not explicitly model choices that may be required for specifying visual mappings, *independent of the data*. These are typically design parameters: the transparency of dots in a dot plot, the number of bins in a histogram, or the bandwidth of a density plot. A central observation from our current work is that different αs require different settings for these parameters. In other words, we present evidence that appropriately setting parameters requires not only appropriate data-driven algorithms, but an understanding of the desired transformations that the viewer should be able to visually distinguish.

If we model these different parameter settings as different visualization mappings (analogous to how we handle different samples from a population), then Figure 3 illustrates how we have not many ω mappings to assess, but we ought to consider the consequences of an *adversarial* setting; are there visualization mappings in which it is possible to *hide* specific flaws in the data-generation process by making particularly bad parameter choices?

4 Adversarial Visualizations

Huff's "How To Lie With Statistics" [24] explores many ways in which statistics can be manipulated to convey misleading impression of the data. More recently, Correll & Heer [13] propose the metaphor of "Black Hat Visualization:" that the designer of a visualization can act as a "man-in-the-middle" attacker between a dataset and the analyst who wishes to understand the data. Our work here studies the possibility



(a)

(b)

Fig. 4: A synthetic adversarial visualization. We added a "flaw" (in this case, 10 sample points all with the same value) to a dataset. Visualizations in the left column are of the original distribution; those on the right are of the flawed distribution. By setting design parameters adversarially (as in Fig. 4a), we can hide this data flaw. Conversely, we can set the parameters to highlight the differences (Fig. 4b).

that a designer can create visualizations that seem to accurately convey data, but in fact hide or obscure important features or flaws, i.e., the visualizations suffers from confusers, in the AVD sense. It is our contention that common univariate visualization methods are at risk of having significant ambiguities, either from the uncritical acceptance of default parameter settings, or, with the possibility of an adversarial visualization creator, from intentionally malicious parameter settings.

To explore the adversarial potential of visualizations of distributions, we created a web-based tool for automatically creating confusers via adversarial setting of design parameters. That is, given an initial set of samples, we introduce an α (a data flaws such as noise, mean shifts, or gaps). We then exhaustively search through a space of "plausible" design parameters assuming a data range of $\{0,1\}$: from 3 to 50 histogram bins, KDE bandwidths of 0.1 to 0.25, and dot plot mark radii of 10 to 25 pixels crossed with mark opacities of 0.2 to 1.0. We report the parameter settings that minimize the average per-pixel CIELAB color difference between the visualization of the original and the visualization of the flawed distribution (an imperfect proxy for ω, $\hat{\omega}$). The smaller this difference, the worse the confuser, and so the more successful the "attack."

Javascript code for creating your own adversarial visualizations using the p5.js framework is available at `https://github.com/AlgebraicVis/SanityCheck` and the supplemental material. Echoing the prior literature, we found that dot plots with high opacity could successfully "cover up" many data flaws. Similarly, small numbers of histogram bins, and wide bandwidths, can result in plots that are virtually identical despite large changes in the underlying distributions.

Fig. 4 shows an example of one such attack. In this case, our original dataset was 50 samples from a Gaussian. We then introduced a mode of 10 additional points with the exact same value within the IQR of the samples. By choosing the design parameters (in this case, the number of bins in a histogram, size and opacity of points in a dot plot, and the bandwidth of a KDE) adversarially, we can almost entirely hide the spike (Fig. 4a). Adding a mode to the data distribution becomes a confuser: for histograms, the spike occurs in the modal bin, causing the other bins to renormalize their heights but otherwise keep a similar shape. For dot plots, small points with maximum opacity hide the new mode among other, sparser sample points. For the density plot, the overlarge bandwidth smooths the mode into the rest of the points. Fig. 4b shows how different parameter settings remove the confuser.

Pixel difference is not always a reasonable proxy for human judgments of similarity in visualization, which focuses on larger-scale visual structures [36]. We present these examples to show that the visual signature of data flaws can be quite subtle (or can be made to be subtle) while still producing visualizations that appear reasonable. Our perceptual study attempts to address these factors directly, while also measuring how robust different visualizations are to these sorts of attacks.

5 Evaluation

For visualizations to work as sanity checks, data flaws must be reliably visually detectable. This in turn means that the visual signatures associated with the flaws must be prominent enough to be visible (to avoid confusers), and robust enough to not be mistaken for visual changes due to different samplings of the data without flaws (to avoid hallucinators). We therefore had the following research questions:

1. How good is the general audience at detecting data flaws in standard visualizations of distributions?

2. Do certain visualizations result in more reliable detectability (and so work better for sanity checks) than others?

3. How sensitive are these visualizations to the design parameters required for their construction?

To answer these questions, we performed a crowdsourced experiment to evaluate how detectable different data flaws are amongst different visualizations, and how robust this detectability is amongst different design parameter settings. Our objective was not to fully map out the space of detectability among features, visualizations, and parameters of interest. Rather, we seek to give preliminary empirical support for our claim that data features can have characteristically different visual impacts on different visualization types, and that the design parameters of these visualizations, even within reasonable ranges, can make these visual signatures more or less prominent.

Experimental materials, including data tables and stimuli generation code, are available at `https://github.com/AlgebraicVis/SanityCheck` and the supplemental material.

5.1 Methods

5.1.1 Lineup Protocol

Prior graphical perception studies, such as Harrison et al. [20] and Szafir [46], measure the signal detection power of different visualizations through binary forced choice tasks of just noticeable differences in signal intensity. Other work assessing the perceived similarity between visualizations exist, often through qualitative metrics such as Likert scales (e.g., Demiralp et al. [15] and Correll & Gleicher [12]. In order to address our research questions within the AVD framework, we required an experimental protocol with aspects of both signal detection and similarity judgment. We therefore adopted the "visual lineups"

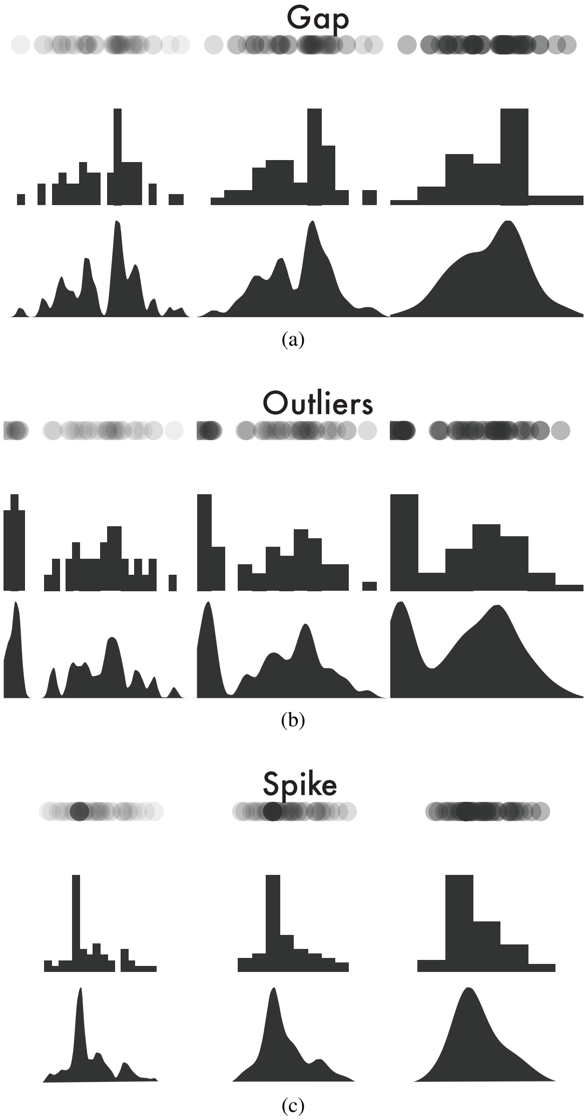


Fig. 5: Examples of our data flaws we tested, and their visual signatures across dot plots, histograms, and density plots. From left to right are examples of the settings of our design parameters: increasing mark opacity, decreasing the number of bins, and increasing kernel bandwidth, respectively.

protocol. The task, as laid out by Wickham et al. [49] is to create a "police lineup" of n visualizations. One of these visualizations contains the actual data "culprit" (e.g., the dataset with the signal present). The other $n-1$ visualizations contain "innocent" data generated under some distribution governed by a null hypothesis. The task is then to identity the culprit. Under the null hypothesis, the probability of selecting the culprit by chance (the false positive rate) is $\frac{1}{n}$. These lineups can therefore be used as a proxy measure of statistical power.

Hofmann et al. [23], as well as Vanderplas et al. [48], have used the lineups protocol to determine the varying signal-detection power of different graphical designs; we use it for a similar purpose here. Beecham et al. [7] and Klippel et al. [31] similarly use a visual lineups to assess the impact of different data properties on visual judgments in maps. The lineup protocol seems to be a reasonably proxy for sanity checking in EDA, as it entails a visual signal detection task in the context of sampling error, as per our formalization in Fig. 2.

In our case, our "innocent" data were random draws from a Gaussian, and the "guilty" data were random draws with some additional data flaw added. For instance, in Fig. 1, the chart on the bottom left of the lineup is "guilty." Each trial of our experiment was a different lineup, with 19 innocent charts, and 1 guilty chart with a flaw. We used 3 different flaw types (Spike, Gap, or Outlier, described below) with 3 different flaw magnitudes (10, 15, or 20 flawed sample points out of 50 total points), across 3 different chart types (dot plot, histogram, or density plot), and 3 different parameters settings (3 different levels of mark opacity, bin count, or kernel size for our 3 chart types, described below), for a resulting $3 \times 3 \times 3 \times 3$ within subjects factorial design, for a total of 81 stimuli.

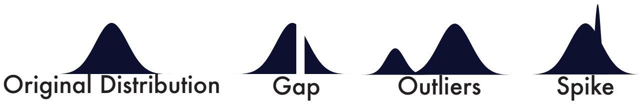


Fig. 6: A representation of the αs that we tested in our study, representing three classes of potential data quality issues. Fig. 5 shows how these flaws might appear across the visualizations we tested in our study.

For training and engagement check purposes, we also included at least one example of each combination of visualization and flaw type (so 9 additional stimuli), with a flaw size of 25 abnormal points, and with parameters we hypothesized to be favorable for detection. These trials were excluded from analysis.

Each visualization consists of 50 samples from a Gaussian distribution with a mean of 0.5 and variance 0.15. We clamped samples to the interval $\{0,1\}$. We generated the resulting visualizations as 150×100 pixel svg images using D3 [9].

5.1.2 Flaw Types

While there are many potential data flaws [29], we focused on three: gaps, spikes, and outliers (Fig. 6). We chose these flaws because they:

- Have a univariate visual signature. That is, the data quality issue can be identified from visual inspection alone. Errors in aggregation or correlation may not be visible from a visualization of a univariate distribution.

- Do not rely on specific semantic information about the domain. For instance, a negative number would be indicative of a data quality issue if the data were framed as age values of people, but would not be if the value were profit/loss information. Crowd-sourced studies are not always reliable for measuring effects based on semantic framing [16].

- Have comparable measures of severity, in terms of number of abnormal points. Other data quality issues may have only two states: present or absent, and so it may be impossible to adjust to identify levels or severity or detectability.

We generated flawed datasets of $50-n$ samples and n abnormal points via the following strategies:

- **Gaps**: We randomly sampled $50+n$ points from the null distribution, and randomly chose a value uniformly from $[Q_1,Q_3]$. We removed the n closest points from this location. This results in an irregularly sized region of missing values somewhere in the middle of the distribution.

- **Outliers**: We randomly sampled $50-n$ points from the null distribution. We measured the post-hoc quartiles of these samples. We then placed n points randomly in either the interval $[0, Q_1 - 1.5{*}\text{IQR}]$ or $[Q_3 + 1.5{*}\text{IQR}, 1]$, whichever was further from the original sample mean. This results in a "clump" of extreme values on one end of the distribution.

- **Spikes**: We randomly sampled $50-n$ points from the null distribution. We measured the post-hoc quartiles of the samples, and randomly chose a value uniformly from $[Q_1,Q_3]$. We added n points with *exactly* this value to the sample. This results in a "spiky" mode somewhere in the middle of the distribution.

5.1.3 Visualization Types

As mentioned in §2, a full analysis of univariate types is out of scope for this paper. We instead focused on three visualizations of distributions that are common in applications such as R, Tableau, VegaLite, and other statistical packages: dot plots, histograms, and density plots. Wilkinson [50] used these visualizations as benchmarks for proposing new visualizations of distributions, as they scale to arbitrary many points without requiring the parametric assumptions of other visualization types (such as the presumed unimodality of box plots). Consult Ibrekk & Granger [25] for an empirical analysis of the legibility of other similar visualizations of distributions with these properties.

The parameters for the visualizations were generated based on scalar multiples of the observed defaults from prior work, based on the idealized sample (which would have $\bar{x} = 0.5$ and $S_x = 0.15$):

- **Dot plot**: We hypothesized that the 0.7 default opacity of Vega-Lite (and certainly the 1.0 default of other VA tools) would result in limited dynamic range in opacity in our dot plots. In piloting, opacities of 0.7 with 50 points produced nearly solid black dot plots, and so we used opacities of $\{0.35, 0.175, 0.0875\}$ in order to avoid floor effects. While the size of marks is also a parameter that can obscure data flaws (and, indeed, it is one of the factors in our attack in §4), we kept a relatively large constant mark radius of 10 pixels both to limit the amount of tested factors, and to create situations in which opacity would result in large visual differences between plots.

- **Histogram**: Sturges' rule would provide 7 histogram bins. We hypothesized that this would result in too few bins, and so created we histograms with $\{7, 14, 28\}$ bins.

- **Density Plot**: We hypothesized that the bandwidth of 0.07 provided by Silverman's rule would result in oversmoothing, and so used bandwidths of $\{0.07, 0.035, 0.0175\}$.

Fig. 5 shows our data flaws, and how they might appear across the different design parameters of our visualizations.

5.1.4 Ecological Validity

There are several significant differences between this task and real-world sanity checks that should promote caution when directly applying our results. In real-world sanity checks, the analyst is unlikely to have access to multiple draws from the same sample, but, rather, one unique dataset. The lineups here complicate flaw detection (in that there are many candidates to examine for flaws), but also might assist in discarding false positives (in that the participant is exposed to the sampling variability first-hand, and so may be better able to distinguish between sampling error and other sources of error). Our forced-choice design also alerts people to the existence of a flaw, and gives no option for participants to refrain from guessing. This prevents the precise identification of hallucinators (false negatives).

In the instructions, the participants were given only a very simple prior about the shape of the distribution: "most of the charts will have the most amount of data in the middle of the chart, and gradually fewer and fewer data points the further from the center of the chart." Beyond this instruction, we did not give any context or fictional framing of our data, as these narrative framings can complicate crowdsourced studies [16]. Real-world analysts are likely to have stronger conceptual models about their data, including knowledge about data format (ages are non-negative integers that are rarely triple digits, for instance), and perhaps even a strong prior about the data distribution (word frequency in texts generally follows Zipf's law, for instance).

We selected 50 points as our data size as a tradeoff between sampling variability and the legibility of individual samples. In practice, the number of points will determine both the legibility of data flaws (1 outlier among a million points may not be visible, for instance) and the design parameters of the visualization (a dot plot with a million points may require differing opacity levels than one with only a dozen).

Lastly, for each trial, the participants were informed of the precise type of flaw they were meant to detect. The full space of dirty data features that an analyst might care about is immense, and a real-world sanity check would entail a simultaneous search across this entire space, rather than a directed search for one specific type of abnormality. While we acknowledge these ecological differences, we argue that the task captures important aspects of sanity checking as a signal detection task.

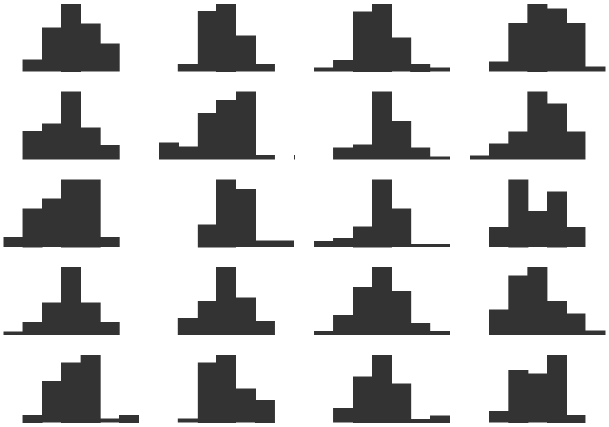


Fig. 7: An example lineup. 19 charts are innocent random samples from a Gaussian, one contains a gap where 15 contiguous points have been removed. When there are only a few histogram bins, these bins are unlikely to match exactly with the gap; it can be hard to distinguish true gaps (which will appear as shorter than expected bars) from variation due to sampling error. That is, the coarse binning functions as a confuser from an AVD perspective. Participant accuracy at gap-detection for histograms with 7 bins was 7%, compared to a chance rate of 5%. See §5.3.4 for the correct answer.

5.2 Hypotheses

Based on our testing concerning the visual properties of visualizations of distributions, we had the following major hypotheses:

- **As the number of flawed points increases, accuracy will increase**. Our model for this task was signal detection. As such, we expected that the strength of the signal (in terms of the number of abnormal points) would result in higher detectability.

- **No one visualization would dominate for all flaw types.** That is, we expected an interaction effect between participant accuracy and visualization type, with no single visualization having consistently higher accuracy. Dot plots highlight individual samples, whereas histograms and density plots highlight the overall shape of the distribution. We expected these differing affordances to have different impacts.

- **More liberal parameter settings will result in increased accuracy**. We define "conservative" parameter settings as low numbers of bins in histograms, high opacity points in dot plots, and large bandwidths in KDEs. By setting these parameters more liberally, we hypothesized that flaw detection would be easier.

5.3 Results

We report our effect sizes using bootstrapped confidence intervals of 90% trimmed means, as per Cleveland & McGill [10]. Trimmed means violate sampling assumptions for standard null-hypothesis tests, however, and so those tests (such as ANOVAs) are reported based on standard means.

5.3.1 Participants

We recruited our participants using `Prolific.ac`. Prolific is a crowdworking platform focused on deploying online studies. Results from Prolific are comparable to those of other crowdwork plaforms [38] such as Amazon's Mechanical Turk. Turk, in turn, has results that are comparable to in-person laboratory studies for graphical perception tasks [21]. We limited our participants to those between 18-65 years of

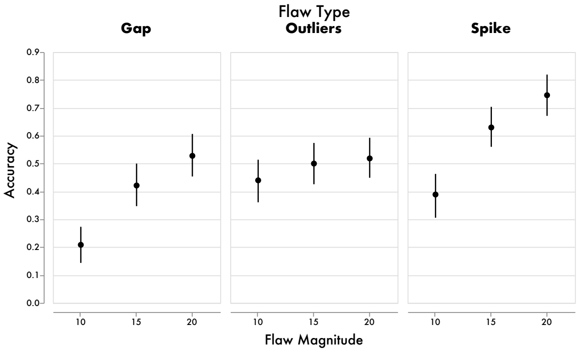


Fig. 8: Accuracy at identifying which visualization contained a specific data flaw, given the size of that flaw in turns of abnormal points (out of 50 total points). Confidence intervals represent 95% bootstrapped C.I.s of the trimmed mean.

age, and with normal or corrected to normal vision. Based on internal piloting, we rewarded participants with $4, for a target compensation rate of $8/hour. Participants, on average, completed the task in 22 minutes ($\sigma_t = 12$ min.), for a post-hoc rate of $10.91/hour.

We recruited 32 participants, 21 men, 11 women, $\mu_{age} = 28.5$, $\sigma_{age} = 8.2$. 26 had at least some college education, of which a further 7 had graduate degrees. We solicited the participants' self-reported familiarity with charts and graphs using a 5-point Likert scale, with 1 being least familiar, and 5 being most familiar. The average familiarity was 2.62, with no participants reporting a familiarity of 5. 2 of our participants did not answer any of the training stimuli correctly, and so their responses were excluded from analysis.

5.3.2 Signal Detection

Chance at this task was $1/20 = 0.05\%$. Across all conditions, participant accuracy was 48.6% (95% bootstrapped c.i. $[46.1\%, 51.2\%]$), significantly higher than chance, although there was significant variation in performance across participants (overall accuracies ranging from 11% for the worst performing participant to 85% for the best performing participant). The average response time from signaling that the participants were ready, to confirming their selected choice, was 10.9s. Not all flaws were equally detectable, however. A Factorial ANOVA found a significant effect of flaw type on accuracy $F(2,58) = 4.7$, $p = 0.013$, as well flaw size in terms of number of affected points ($F(2,58) = 58$, $p < 0.001$). Their interaction was also a significant effect ($F(4,116) = 6.3$, $p < 0.001$).

In general, spikes were the easiest to detect, with an accuracy of 58.8% $[54.4\%, 63.1\%]$, followed by outliers at 48.6% $[44.2\%, 53.0\%]$, with gaps being the hardest to detect at 38.6% $[34.4\%, 42.8\%]$ accuracy. A post-hoc pairwise Bonferroni-corrected t-test showed that all three flaw types were significantly different from each other. It was generally the case that increasing the size of the flaw significantly increased its detectability: a post-hoc pairwise Bonferroni-corrected t-test of the interaction of flaw magnitude and flaw type on accuracy found that gap and spike detection was significantly more accurate with flaw sizes of 20 and 15 versus those of size 10. However, the number of outliers did not significantly affect their detectability. Fig. 8 shows these results in more detail. In general, these results partially supported our first hypothesis: **Participants were better able to detect flaws as these flaws were larger**, with the exception of outliers.

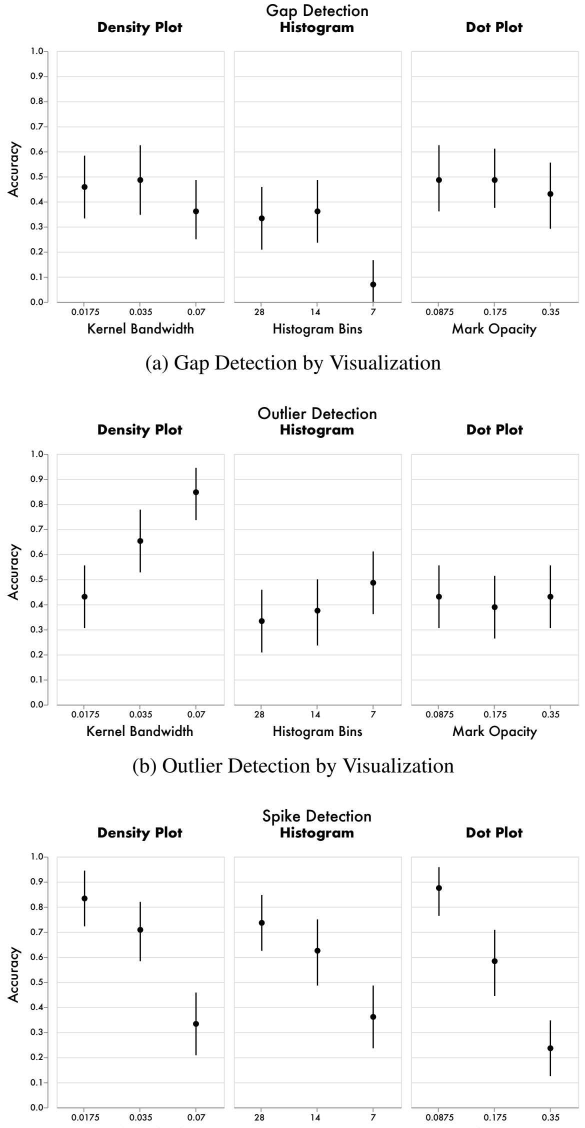
5.3.3 Visualization Performance

Not all visualizations were equally useful for detecting flaws. A Factorial ANOVA found a significant affect of visualization type on accuracy ($F(2,58) = 10.5$, $p < 0.001$), as well as a significant interaction effect between visualization type and flaw type ($F(4,116) = 4.5$, $p = 0.002$).

Across all conditions, density plots were the most accurate, 56.8% $[52.3\%, 61.2\%]$, followed by dot plots, 48.3% $[44.1\%, 52.5\%]$, with histograms being the least accurate, 40.9% $[36.5\%, 45.3\%]$. A post-hoc pairwise Bonferroni-corrected t-test found that all visualization types were significantly different from each other.

Despite this ranking, our results partially support our second hypothesis: **no single visualization was significantly better for detecting every type of data quality issue**. A post-hoc pairwise Bonferroni-corrected t-test of the interaction between visualization type and flaw type found that density plots were significantly better than other charts for outlier detection (64.3% $[57.1\%, 71.6\%]$ vs. 39.8% $[32.1\%, 47.4\%]$ for histograms and 41.7% $[34.4\%, 49.0\%]$ for dot plots), and that histograms were significantly worse than other charts for detecting gaps (25.5% $[18.7\%, 32.2\%]$ vs. 43.5% $[35.9\%, 51.1\%]$ for density plots and 46.8% $[39.0\%, 54.5\%]$ for dot plots). All other charts were comparable within flaw types.

5.3.4 Parameter Settings



(a) Gap Detection by Visualization

(b) Outlier Detection by Visualization

(c) Spike Detection by Visualization

Fig. 9: Performance at our flaw detection task across designs and their parameters. Each rightmost column across designs represents a default from Silverman's Rule, Sturges' Rule, and $\frac{1}{2}$ VegaLite, respectively. With the exception of the outlier detection task, deviating from these defaults generally resulted in better performance. Confidence intervals represent 95% bootstrapped C.I.s of the trimmed mean.

Accuracy at detecting flaws was dependent on the settings of the design parameters of the visualizations. In the most extreme case, doubling the amount of bins in a histogram from the 7 recommended by Sturges' rule to 14 resulted in a nearly 5-fold increase accuracy at detecting gaps (from 7%, near chance, to 36%). Fig. 7 shows an example of this extreme case. In that figure, the guilty visualization is the right-most column, third from the top.

Our results only partially support our third hypothesis: **liberal parameters often had higher accuracy than more conservative ones**. A Factorial ANOVA of accuracy found a significant interaction effect between the flaw type and the conservativeness of the parameter setting ($F(4,116) = 26$, $p < 0.001$). We conducted a post-hoc pairwise Bonferroni-corrected t-test of the interaction between parameter settings and flaw type. For spikes, we found that all three levels of parameter settings were significantly different from each other, with more liberal settings outperforming more conservative settings. Whereas for gaps, all three settings were not significantly different. For outlier detection, the most conservative parameter setting significantly *overperformed* the most liberal ones. An analysis of this result shows that it is mostly due to density plots: since the outliers were sufficiently far from the mode to not be "smoothed away" by large bandwidth kernels, the increased kernel bandwidth had the effect of making the outliers more prominent.

An analysis of our per-visualization results show that the aggregate group differences were driven by several particularly beneficial or harmful parameter settings, in keeping with our adversarial results from §4: small bandwidths KDEs make spikes especially prominent, as the large number of overlapping points in a small region create a large visual spike that is not smoothed into the neighboring points. Histograms with small numbers of bins make the detection of spikes and gaps difficult, as the spike causes renormalization (as in Fig. 4), and the bin containing the gap will contain many neighboring values, resulting in a subtle visual change rather than a sudden drop-off in density. Lastly, high-opacity dot plots make internal modes indistinguishable from overplotted regions. Fig. 9 shows these results, pivoting on detection type, task, and parameter setting.

6 Discussion

Our results identify some potential concerns with the use of visualizations as sanity checks. Overall, it appears that the detection of data flaws from visualizations is by no means reliable, or neatly disentangled from differences caused by sampling error: even spikes, the flaw with the highest overall detection rate, were correctly identified only 59% of the time. Real world datasets, with more subtle data flaws, larger sample sizes, and more complex sources of error, should induce even more skepticism about the ability of standard visualizations to reliably surface data flaws. We should expect a larger class of confusers and hallucinators among the wider class of visualizations and flaw types. Designers of visual analytics systems, especially those for EDA or data cleaning, should conduct their own analyses of the AVD properties of their visualizations to assess the robustness of competing designs.

Our results also suggest that there is no single visualization design, or parameter setting, that will make all potential data quality issues equally visible. While density plots were the most robust visualization we tested (across all flaws, they were either the visualization with the highest performance, or not significantly different from the visualization with the highest performance), density plots also showed wide variability in their performance based on their bandwidth. Disadvantageous or adversarial setting of design parameters can erase the performance benefits of any particular visualization design.

Lastly, our results suggest that, in many cases, more liberal settings of things like histogram bin size and mark opacity will result in better performance. In the following section, we speculate on the visual impact of these features across each visualization type we tested, and their implications for designing for sanity checking.

6.1 Density Plots

We were in general surprised by the relatively high performance of density plots across conditions, as the specifics of their design is based on a concept, KDE, that requires a reasonable amount of statistical background to interpret. Their performance here indicates that they may be justified for inclusion among the standard arsenal of distribution visualization tools in visual analytics systems, despite their sensitivity to their underlying parameters.

As discussed in prior work, there appears to be a significant cost for oversmoothing in density plots with respect to detecting gaps and modes. Oversmoothed modes are "blended" into the regular distribution. Oversmoothed gaps make the case of *no* data ambiguous from the case *less* data, especially if they occur in regions with large amounts of data on either side of the gap. Our results show a monotonic decrease in performance for these tasks as bandwidths get larger.

However, we observed a positive benefit for large bandwidths in outlier detection. On reflection, outliers are large numbers of points in sparse regions: a large bandwidth would make these outliers "wider" and more visually prominent. Large bandwidths also help disambiguate the small number of extreme points that would occur due to sampling error (which would be smoothed into the larger shape of the distribution) from a cluster of systematically outlying points (which stick out).

Therefore, while we recommend that designers consider using density plots in wider applications, we suggest that they be mindful of the data quality flaws they expect their analysts to encounter, and consider that rules of thumb such as Silverman's rule may be too conservative to show much in the way of the internal structure of distributions.

6.2 Histograms

We were also surprised by the relatively poor performance of histograms; in no condition were they significantly better than other visualization types for detecting flaws, and in one case (gap detection), they were significantly *worse* than their competitors. Histograms are a standard tool for visualizing distributions, and are the default per-field visualization in commercial data prep tools such as Trifacta's Wrangler and Tableau Prep. Histograms do have some advantages in terms of the scalability of their design (affording the visualization of arbitrary numbers of data points in a constrained visual space with no potential for overplotting), but our results indicate that they, in themselves, may only be weak evidence for the presence or absence of a flaw in the data.

Expanding on existing results, we show that there are severe performance costs to underestimating the number of histogram bins. If there are too few bins, then missing data can be small enough to be drowned out by dense neighbors. As with density plots, this underestimation also causes an ambiguity between *no* data and *less* data, a signal that can be confused with variability due to sampling error. Similarly, histograms are often normalized with respect to their maximum observed density. Therefore, extraneous modes that are too close to existing modes can just force a renormalization of all the other bins, while leaving the modal bin unchanged. This "renormalization bias" can obscure important patterns in both univariate and spatial visualizations [14]. Large bins, however, can group all outliers into a single bin of locally high density. These large outlying bins are more prominent than if outliers cross multiples bins. They also reduce some of the ambiguity in whether there are enough extreme points to indicate a data quality issue, or if they are artifacts of sampling error.

In general, we do not recommend the use of Sturges' rule for creating sanity checking histograms [42]. Other simple rules of thumbs, such as the Freedman-Diaconis rule [18], are more liberal (it would generate 13 bins for our idealized distribution, instead of the 7 of Sturges' rule). Even so, with extremely liberal bin settings, other visualization types that do not discretely aggregate data are likely to expose a wider class of data flaws than histograms.

6.3 Dot plots

We expected strong performance from dot plots which, under reasonable settings, have direct and unambiguous visual signatures for each one of our data flaws. For instance, modes appear as solid circles surrounded by less opaque neighbors, and gaps appear as continuous regions with no marks whatsoever (see Fig. 5). We expected most performance issues to stem from overplotting.

Unlike histogram binning, which has been studied for decades, work on automatically setting the opacity of points in a scatterplots and dot plots is relatively recent [33, 35]. As such, there are few heuristic rules for setting mark opacity. Our results, while coarse, do not suggest any particular local maximum of performance: most opacities, so long as they are low enough to reduce the effects of overplotting, will still afford many forms of sanity checking. Not just the opacity, but the size of the marks, and the density of the data, will contribute to the severity of overplotting.

The relative robustness of dot plots could be due to the fact that, for feasibility reasons, our list of viable parameters for mark opacity began relatively low (half of VegaLite's default mark opacity of 0.7). We piloted with different maximums (including 0.7 and 1.0), but encountered floor effects, as most of the dot plots were very close to solid black lines of overplotted marks. Therefore, we still recommend that designers be mindful of dot plot opacity, and generally more liberal in setting defaults than those of alternative analytics tools. Our results indicate that, unlike the other visualization types where there were performance benefits on both extremes of the parameter scale, data flaws are visible even with very low mark opacity.

When employing dot plots for sanity checking purposes, we recommend that designers, by default, employ some method of ameliorating overplotting, either in terms of mark opacity, jittering of points, or reduction of mark size. Constant opacity values, that do not take into consideration the number of modality of points, may not be sufficient measures to produce usable dot plots by default.

7 Limitations & Future Directions

7.1 Limitations

We explored a narrow and coarse range of design parameters in our lineup study. It is not within the scope of this paper to generate full models of the impact of design parameters (such as histogram bin size or KDE bandwidth) on different flaw detection tasks. As noted in §2, smarter strategies for automatically setting parameters may avoid some confusers created by the default settings we studied. Our goal, however, was not to derive new or test all possible strategies, but to examine how design parameters are set in practice, and to explore the effects of these settings on data sanity checking. We contend that analysts may not be aware of the existence of more complex parameter-setting rules, or may not feel the need to adjust the initial univariate plots. We hope our work prompts designers of future visual analytics systems to think more carefully about how these defaults are set, and how the system can encourage user interaction with these initial plots.

We also limited the visualizations we tested to a small class of well-known exemplars. The computation of KDE, for example, can borrow any number of visualization techniques that have been evaluated in the context of either probability distributions [25] or time series visualization [19]. Again, we were focused on sanity checking visualizations as they occur in common visual analytics systems. More complex visualizations (such as Summary Box Plots [39]) may directly encode features relevant for data quality, but at the cost of increased visual complexity or training time. Designers should weigh these costs (and the false positive and false negative costs of detecting potential data quality issues) when determining whether or not to support more complex initial views of distributions.

Our experimental task, by forcing participants to make a choice, does not distinguish false positives (they selected a chart with no flaw, but a visual distinction that is a mere result of sampling error) from false negatives (they could not find *any* chart that appeared flawed, and so guessed randomly). Therefore, in the language of AVD, our results cannot distinguish between confusers and hallucinators. We intend to disambiguate these scenarios in future work, but speculate that for data sanity checking, false negatives may be more problematic (we want to avoid using dirty data). EDA tools may therefore value the absence of confusers more than the absence of hallucinators.

Lastly, we used naïve participants with no particular stakes in, or strong semantic connections with, the data. Statistical expertise, visual literacy, or strong priors from the analytical domain, could greatly improve performance at sanity checking. It remains an open research question to quantify how semantic information impacts performance at lower-level graphical perception tasks, and whether higher stakes in decision-making encourages fundamentally different patterns of information-seeking and verification. Similar, the impact of domain knowledge or other semantic information on perceptions of data quality is unexplored in this work. Further ethnographic work may illuminate whether there are visual signatures that function as sanity checks in specific data domains, or among more experienced data scientists.

7.2 Potential Solutions & Future Work

One solution to the issues raised in this paper is to integrate automatic anomaly detection with visualization directly. Tools like Profiler [28] use approaches from data mining to suggest potential data anomalies, and then allow analysts to visually diagnose their source or assess their severity. We recommend a further integration of anomaly detection and visualization: anomaly detection might function as the visual analytics equivalent of "compiler warnings:" by automatically detecting anomalies that are not readily visible in particular visualization types (such as gaps in histograms), a system could prompt interaction or other mixed-initiative solutions (such as a focus+context view inside a particularly suspicious histogram bin). This visualizations+warnings design could accelerate sanity checking when data quality issues will be readily apparent, but encourage the analyst to slow down and interact when issues are less prominent. Likewise, recommender systems such as Voyager [51] could use automatic methods to explicitly guide analysts to closely examine problematic fields in a dataset. We intend to explore how to combine automated anomaly detection with visual analytics, and how to communicate potential data quality issues to analysts, including at stages downstream of sanity checking.

Another class of solutions is to employ hybrid or ensemble visualizations as the default view for data prep contexts. Bean plots [27] (with low opacity interior strips) represent a potential ensemble design with components that, in our study, afforded detectability across the full range of the data flaws we explored. Likewise, hybrid visualizations such as Wilkinson dot plots [50] or beeswarm plots [17], when the dataset is small enough, could afford some of the benefits of both dot plots (in that individual points can be visible) and density plots (in that the overall shape of the distribution can be visible). We intend to explore this space in more detail, and empirically test our supposition that these designs can provide the additive benefits of their components.

Our work has concrete implications not just for designers using standard visualizations in visual analytics contexts, but also for designers creating and evaluating new visualization methods. We encourage designers to not only test the robustness of their design parameters in terms of the quality of their visualizations in normal scenarios, but to specifically consider whether data quality concerns are easily discoverable or detectable across the relevant parameter spaces. Ideally, designers would build their new techniques *defensively*, such that important data quality concerns are difficult to ignore, even across a wide (or perhaps adversarial) range of parameter settings.

7.3 Conclusion

In this work, we examine the capabilities of visualizations to act as sanity checks: simple visualizations that are meant to be used to rapidly confirm that a particular dimension of a dataset is relatively free from flaws. The sanity checking process may be brief, and the analyst may be discouraged from interacting with or otherwise altering the design of these preliminary visualizations. In this setting, we have shown that there is a wide class of visualizations that appear to plausibly summarize data, but make flaws in the data difficult to detect.

Acknowledgments

Scheidegger and Li's work in this project was partially supported by NSF award IIS-1513651 and the Arizona Board of Regents.

References

[1] D3: histogram.js. https://github.com/d3/d3-array/blob/master/src/histogram.js.

[2] R: bandwidth. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/bandwidth.html.

[3] R: hist. http://stat.ethz.ch/R-manual/R-devel/library/graphics/html/hist.html.

[4] Tableau help: Create bins from a continuous measure. https://onlinehelp.tableau.com/current/pro/desktop/en-us/calculations_bins.html.

[5] Vega-lite: init.ts. https://github.com/vega/vega-lite/blob/cebdffb20517947bd6dadb040ffe80f00d5bfcb2/src/compile/mark/init.ts.

[6] F. Anscombe. Graphs in statistical analysis. *The American Statistician*, 27(1):17–21, 1973.

[7] R. Beecham, J. Dykes, W. Meulemans, A. Slingsby, C. Turkay, and J. Wood. Map lineups: effects of spatial structure on graphical inference. *IEEE transactions on visualization and computer graphics*, 23(1):391–400, 2017.

[8] S. Belia, F. Fidler, J. Williams, and G. Cumming. Researchers misunderstand confidence intervals and standard error bars. *Psychological methods*, 10(4):389, 2005.

[9] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011.

[10] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984.

[11] M. Correll and M. Gleicher. Error bars considered harmful: Exploring alternate encodings for mean and error. *IEEE transactions on visualization and computer graphics*, 20(12):2142–2151, 2014.

[12] M. Correll and M. Gleicher. The semantics of sketch: Flexibility in visual query systems for time series data. In *Visual Analytics Science and Technology (VAST), 2016 IEEE Conference on*, pp. 131–140. IEEE, 2016.

[13] M. Correll and J. Heer. Black hat visualization. In *Workshop on Dealing with Cognitive Biases in Visualisations (DECISIVe), IEEE VIS*, 2017.

[14] M. Correll and J. Heer. Surprise! Bayesian weighting for de-biasing thematic maps. *IEEE transactions on visualization and computer graphics*, 23(1):651–660, 2017.

[15] Ç. Demiralp, M. S. Bernstein, and J. Heer. Learning perceptual kernels for visualization design. *IEEE transactions on visualization and computer graphics*, 20(12):1933–1942, 2014.

[16] E. Dimara, A. Bezerianos, and P. Dragicevic. Narratives in crowdsourced evaluation of visualizations: A double-edged sword? In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 5475–5484. ACM, 2017.

[17] A. Eklund. Beeswarm: the bee swarm plot, an alternative to stripchart. *R package version 0.1*, 5, 2012.

[18] D. Freedman and P. Diaconis. On the histogram as a density estimator: L 2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57(4):453–476, 1981.

[19] J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, and P. Isenberg. Evaluation of alternative glyph designs for time series data in a small multiple setting. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 3237–3246. ACM, 2013.

[20] L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking visualizations of correlation using weber's law. *IEEE transactions on visualization and computer graphics*, 20(12):1943–1952, 2014.

[21] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 203–212. ACM, 2010.

[22] J. L. Hintze and R. D. Nelson. Violin plots: a box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998.

[23] H. Hofmann, L. Follett, M. Majumder, and D. Cook. Graphical tests for power comparison of competing designs. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2441–2448, 2012.

[24] D. Huff. *How to lie with statistics*. WW Norton & Company, 2010.

[25] H. Ibrekk and M. G. Morgan. Graphical communication of uncertain quantities to nontechnical people. *Risk analysis*, 7(4):519–529, 1987.

[26] M. C. Jones, J. S. Marron, and S. J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996.

[27] P. Kampstra. Beanplot: A boxplot alternative for visual comparison of distributions. 2008.

[28] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 547–554. ACM, 2012.

[29] W. Kim, B.-J. Choi, E.-K. Hong, S.-K. Kim, and D. Lee. A taxonomy of dirty data. *Data mining and knowledge discovery*, 7(1):81–99, 2003.

[30] G. Kindlmann and C. Scheidegger. An algebraic process for visualization design. *IEEE transactions on visualization and computer graphics*, 20(12):2181–2190, 2014. doi: 10.1109/TVCG.2014.2346325

[31] A. Klippel, F. Hardisty, and R. Li. Interpreting spatial patterns: An inquiry into formal and cognitive aspects of tobler's first law of geography. *Annals of the Association of American Geographers*, 101(5):1011–1031, 2011.

[32] A. Lunzer and A. McNamara. It aint necessarily so: Checking charts for robustness. *IEEE VisWeek Poster Proceedings*, 2014.

[33] J. Matejka, F. Anderson, and G. Fitzmaurice. Dynamic opacity optimization for scatter plots. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 2707–2710. ACM, 2015.

[34] J. Matejka and G. Fitzmaurice. Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 1290–1294. ACM, 2017.

[35] L. Micallef, G. Palmas, A. Oulasvirta, and T. Weinkauf. Towards perceptual optimization of the visual design of scatterplots. *IEEE transactions on visualization and computer graphics*, 23(6):1588–1599, 2017.

[36] A. V. Pandey, J. Krause, C. Felix, J. Boy, and E. Bertini. Towards understanding human similarity perception in the analysis of large sets of scatter plots. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 3659–3669. ACM, 2016.

[37] B. U. Park and J. S. Marron. Comparison of data-driven bandwidth selectors. *Journal of the American Statistical Association*, 85(409):66–72, 1990.

[38] E. Peer, L. Brandimarte, S. Samat, and A. Acquisti. Beyond the turk: Alternative platforms for crowdsourcing behavioral research. *Journal of Experimental Social Psychology*, 70:153–163, 2017.

[39] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson. Visualizing summary statistics and uncertainty. In *Computer Graphics Forum*, vol. 29, pp. 823–832. Wiley Online Library, 2010.

[40] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2017.

[41] D. W. Scott. Kernel density estimators. *Multivariate Density Estimation: Theory, Practice, and Visualization*, pp. 125–193, 2008.

[42] D. W. Scott. Sturges' rule. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(3):303–306, 2009.

[43] S. J. Sheather. Density estimation. *Statistical science*, pp. 588–597, 2004.

[44] S. J. Sheather and M. C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 683–690, 1991.

[45] M. Stone, D. A. Szafir, and V. Setlur. An engineering model for color difference as a function of size. In *Color and Imaging Conference*, vol. 2014, pp. 253–258. Society for Imaging Science and Technology, 2014.

[46] D. A. Szafir. Modeling color difference for visualization design. *IEEE transactions on visualization and computer graphics*, 24(1):392–401, 2018.

[47] J. W. Tukey. *Exploratory data analysis*, vol. 2. Reading, Mass., 1977.

[48] S. VanderPlas and H. Hofmann. Spatial reasoning and data displays. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):459–468, 2016.

[49] H. Wickham, D. Cook, H. Hofmann, and A. Buja. Graphical inference for infovis. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):973–979, 2010.

[50] L. Wilkinson. Dot plots. *The American Statistician*, 53(3):276–281, 1999.

[51] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics*, 22(1):649–658, 2016.