DOI: 10.1111/cgf.13975
Eurographics Conference on Visualization (EuroVis) 2020
M. Gleicher, T. Landesberger von Antburg, and I. Viola
(Guest Editors)

Volume 39 (2020), Number 3

VisuaLint: Sketchy In Situ Annotations of Chart Construction Errors

Aspen K. Hopkins[1], Michael Correll[2], and Arvind Satyanarayan[1]

[1] MIT CSAIL [2] Tableau Research

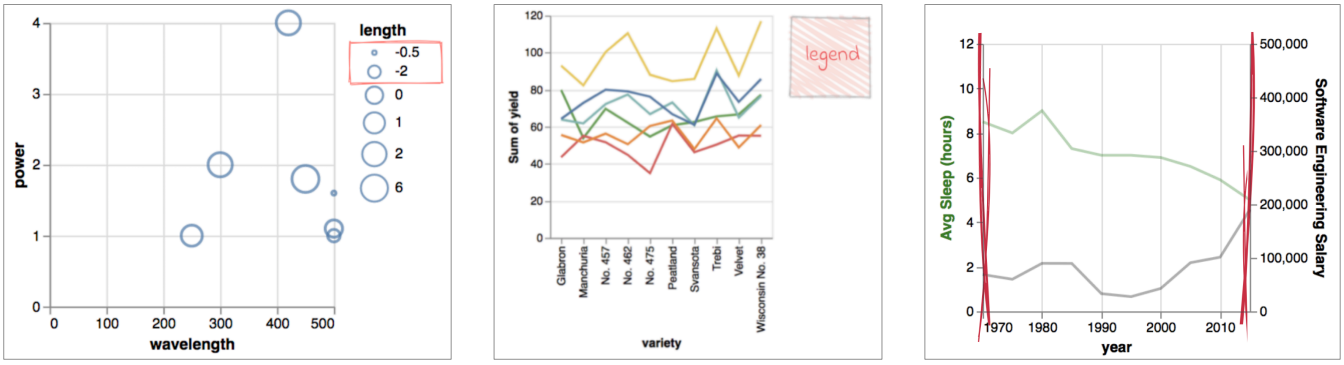

Figure 1: *Examples of VisuaLint for (left-to-right) an inexpressive size encoding, a missing legend, and dual-axes with differing scale rates.*

Abstract

Chart construction errors, such as truncated axes or inexpressive visual encodings, can hinder reading a visualization, or worse, imply misleading facts about the underlying data. These errors can be caught by critical readings of visualizations, but readers must have a high level of data and design literacy and must be paying close attention. To address this issue, we introduce VisuaLint: a technique for surfacing chart construction errors in situ. Inspired by the ubiquitous red wavy underline that indicates spelling mistakes, visualization elements that contain errors (e.g., axes and legends) are sketchily rendered and accompanied by a concise annotation. VisuaLint is unobtrusive — it does not interfere with reading a visualization — and its direct display establishes a close mapping between erroneous elements and the expression of error. We demonstrate five examples of VisualLint and present the results of a crowdsourced evaluation (N = 62) of its efficacy. These results contribute an empirical baseline proficiency for recognizing chart construction errors, and indicate near-universal difficulty in error identification. We find that people more reliably identify chart construction errors after being shown examples of VisuaLint, and prefer more verbose explanations for unfamiliar or less obvious flaws.

CCS Concepts

• ***Human-centered computing*** → ***Visualization techniques;***

1. Introduction

Chart construction errors — mistakes in visual encodings that are agnostic to underlying data semantics such as truncated axes, misaligned baselines, or inexpressive channels — are commonplace. Every week, dozens of real-world examples of such errors are posted to the *r/DataIsUgly* Reddit subcommunity, spanning a variety of domains including quarterly financial reports, renowned newspapers, and scientific publications. It can be tempting to dismiss these issues as "amateur mistakes", but expert data visualization designers are prone to making them as well — for instance, a recent blog post by the visual data journalism team at *The Economist* documented seven examples of such mistakes in their work [Leo19]. At best, these errors result in visualizations that are confusing or difficult to read and interpret. At worst, such mistakes can yield charts that mislead readers and shape discourse about the underlying data in problematic ways. As an example, during a high-profile September 2015 United States congressional hearing, a chart was shown that purported to depict Planned Parenthood dramatically increasing the number of abortions it offered as compared to cancer screenings and preventative services (Fig. 2(left)). This visualization was an instance of a dual-axis chart using drastically different scales; when journalists replotted the data against a common baseline, the trend was muted (Fig. 2(right)).

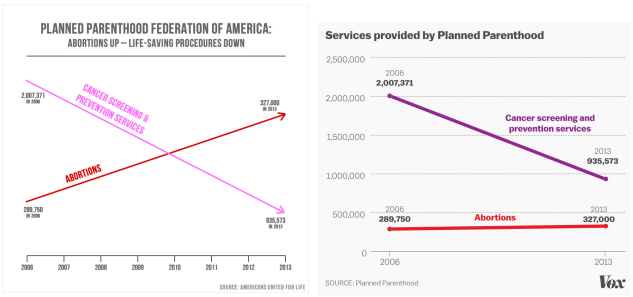

Figure 2: *Planned Parenthood Abortion Rates: Before and After Dual Axes Error is Corrected.*

Identifying chart construction errors is not always straightforward. Dozens, if not hundreds, of rules are necessary to express best practices in visualization design [MK18, MWN*19, Mee17] — an infeasible level of knowledge to expect of casual authors and readers, exacerbated by the ever-evolving nature of visualization best practices [Kos16]. Even for experts that possess a high level of data and design literacy, catching these mistakes is non-trivial and requires critically reading charts with careful attention — an activity made all the more challenging due to the sense of authority and certainty visualizations convey [KHAA16, Kos08]. As a result, surfacing these mistakes occurs in an unstructured and ad hoc fashion, and often relies on experts such as data journalists or data activists who largely focus on a narrow set of high profile domains.

There is a growing research interest in addressing these issues, but existing work has primarily done so via systems-building. For instance, novel systems that automatically detect errors in a visualization [MK18], codify best practices and recommend alternative designs [MWN*19], or learn design principles from corpora of examples [SML*18] have been proposed or developed. While valuable, this focus approaches the problem space from the perspective of visualization *authors* and does not grapple with how visualization *readers* reason about chart construction errors that they might experience in the wild. As a result, user experience and usability concerns associated with error detection (e.g., how might we indicate visualization errors to lay readers, and how well do lay readers understand the error that has occurred) have yet to be explored.

As a way of focusing on not just *what* errors are present, but also how to *surface* them, we present VisuaLint: a novel technique for expressing chart construction errors to a lay audience. VisuaLint provides a general design language that can be adapted and extended for an ever-evolving set of visualization best practices. Erroneous visualization elements (e.g., axes and legends) are sketchily rendered — an approach inspired by Wood et al. [WII*12] wherein visual elements are presented in a "hand-drawn" manner — and can be accompanied by annotations for additional context. For instance, missing legends are highlighted by appending a sketchy, pseudo-legend to the chart; if dual-axes charts used misaligned scales, the axes are replaced with vertical, red wavy lines; and, if bar charts do not begin at a zero baseline, a sketchy arrow and 0-label annotation is overlaid on the y-axis. VisuaLint is designed to be salient yet unobtrusive — errors are highlighted, facilitating identification, but the expression does not interfere with chart reading. Its direct, in situ presentation facilitates a *closeness of mapping* [BBC*01] more analogous to spell check annotations than code linting tools, which typically list errors in a secondary view. By depicting errors as *sketchy* rather than *crisp* elements, VisuaLint helps undermine the assumed authority of visualizations [KHAA16].

We evaluate VisuaLint in two ways. We demonstrate its expressive extent through several example visualizations that cover a range of mistakes including ineffective color choices, inexpressive size encodings, issues with scale and axis baselines and alignment, and missing elements such as legends. To assess its efficacy at surfacing errors, and to solicit users' preferences, we conducted a two phase crowdsourced study: participants were asked to identify chart construction errors before and after being exposed to VisuaLint, and then to rank different expressions of these errors. Through this study, we contribute a empirical baseline for proficiency in recognizing chart construction errors. We show that, without intervention, readers have great difficulty recognizing these errors. Our results also show that people more reliably identify such errors after being exposed to VisuaLint, and suggest that people appreciate greater levels of guidance especially in cases where the chart construction errors are complex or esoteric.

Experimental materials, including stimuli and data tables, are hosted on OSF: `https://osf.io/jwbn2/`.

2. Related Work

Improperly designed visualizations can have dramatic impacts on how data are (or are not) interpreted. Moreover, certain sets of visualizations have been labelled as "deceptive," with measurable biases in how people interpret data [PRS*15].

Despite this potential for harm or misuse, guidance on avoiding visualization "pitfalls" [BE15] is often absent in visualization tools. When best practices or design issues are codified, they are often presented to the user implicitly: for instance, through "smart defaults" in languages like Vega-Lite [SMWH16], or as constraints in recommendation systems like Draco [MWN*19]. While helpful, these implicit approaches can fail when users deviate from expected use cases. More importantly, they provide little pedagogical value as the choice of default values are rarely explained. There is a dearth of work on *in situ* feedback for visualization design, and on validating and verifying visualizations [KS08].

Given the severity and costliness of errors in data analysis, there is an emerging scholarship on using automated or semi-automated methods to detect potential flaws in analyses. For instance, Barowy et al. [BGB14, BBZ18] investigate systems for automatically detecting errors in spreadsheets. More specialized tests have been proposed for analysis concerns such as Simpson's paradox [GBK17], the "drill down fallacy" [LDH*19], and the multiple comparisons problem [BDSK*17] (specifically as it emerges in visual analytics contexts [ZZZK18]). Similar to our work, McNutt et al. [MKC20] investigate a general class of visualization "mirages," and suggest testing regimes to disclose these chart errors to readers.

A related concern above and beyond analytical correctness is how to build skepticism and encourage critique of visualizations. The clean design and inherent rhetorical force of visualizations lend them implicit authority [KHAA16, Kos08]. Colloquial attitudes towards visualizations can include a disregard of the provenance of

data or potential biases of designers in the belief that charts are objective representations of truth [PAEE19]. Compounding issues of cultural norm and limited data literacy discourage critical, intentional chart reading and perpetuate poor visualization practices.

To address both of these concerns, we borrow the metaphor of a "linter," a concept from static code analysis in which the source code of a program is checked for errors in syntax, style, or even simply constructs that are often misused [Lou06]. A linter presents the user with a list of violations of linting rules, which the user can either fix or ignore. Hynes et al. [HST17] propose a data linter for "sanity checking" datasets for ML purposes. Closer to our work, McNutt and Kindlmann [MK18], drawing on existing visualization best practices, propose a linter for visualizations. This prior work, however, has primarily focused on linting *systems* — using automated or semi-automated ways of detecting visualization errors — and there has been little commensurate work studying how to *present* the results of a visualization linter in a usable fashion to readers. In response, our work primarily focuses on this latter issue. We draw on a key affordance of visualizations: they afford the presentation of *in situ* information about violations of linting rules. Thus, we generate *visual lints* that overlay the chart, and direct the viewer to specific sections in which a rule violation was detected, using the same visual language as the chart itself. Our approach is inspired by prior work by Hoffswell et al. [HSH18] who show that users are able to more quickly and accurately debug code when editors are annotated with in situ visualizations of program state.

3. The Design of VisuaLint

We designed four lint motifs through an iterative process informed by prior work on perceptual studies, visualization best practices [Tuf01] and linting [Mee17, MK18]. We evaluated each motif through informal, unstructured interviews with users spanning a range of data expertise.

3.1. Design Process

Although the research community has increasingly been devoting attention to the issue of visualization errors, a canonical list of such errors does not yet exist. As a result, we began our design process by compiling a list of data visualization best practices and associated errors, starting with a review of the literature on perceptual studies, visualization systems [MWN*19, SMWH16, MK18], and widely-read, non-academic material such as Tufte's *The Visual Display of Quantitative Information* [Tuf01]. To ensure our list of errors also reflected real-world practice, we additionally collated discussions of visualization best practices found in online forums (e.g., VisGuides [DAREA*18], and Reddit's *r/DataIsUgly*) as well as examples drawn from current affairs and the media (e.g., congressional hearings as in Fig. 2 or blog posts from journalists [Leo19]).

Through subsequent discussions, we crafted several methods of categorizing errors. We hoped that, through this process, we would uncover commonalities between errors that would in turn inspire VisuaLint's design. On reflection, we noticed a shared feature in our categorizations—the separation between data-centric errors, those grounded in data semantics, and construction errors. This distinction guided our evaluation design, as construction errors are visually recognizable to a reader regardless of prior knowledge of the underlying data. By focusing on chart construction errors, we could examine lay comprehension of visualizations.

We conducted informal, unstructured interviews with eight individuals (three men and five women) with varying levels of data literacy and familiarity with visualizations. Four participants were PhD students in data-related fields, three had taken or were in an undergraduate statistics course, and one had completed a high school degree but had no exposure to data-related topics. Ages ranged from 21-30 years. Interviews lasted 35–45 five minutes, and covered each design motif and their various iterations.

We presented participants with a variety of errors for each motif, and posed open-ended questions such as *"what do you think is being indicated on the chart?"*, *"why do you think that?"*, and *"how might this design be improved?"*. After all motifs were presented, a final question solicited overall preferences: *"of these different designs, which makes the most sense, and why?"*. We documented these responses, which helped us synthesis our design goals (§ 3.2) as well as potential refinements to the motifs.

3.2. Design Goals

Through our successive rounds of formative evaluations, we synthesized the following design goals for VisuaLint:

- **G1 Salient but Unobtrusive.** This goal balances the need of noticeably surfacing errors with maintaining the legibility of the underlying visualization. Instances of VisuaLint must be immediately apparent to readers, but must not interfere with reading the visualization. For example, readers should not confuse VisuaLint elements as being a part of the original chart. This goal seeks to both reduce the amount of attention a reader must pay to identify errors, but also the cognitive burden associated with interpreting and distinguishing VisuaLint from the visualized data.
- **G2 Direct.** Rather than listing errors in a secondary view, akin to many code linting tools, user feedback favored an approach more analogous to spell check's red wavy line. VisuaLint is displayed *in situ*, either overlaying or replacing erroneous elements. By aligning chart elements with error expression, VisuaLint establishes a *closeness of mapping* [BBC*01] that facilitates error awareness.
- **G3 Composable.** Unlike spell check, where there is only one reason a word may be underlined (it is spelled incorrectly!), there may be a multitude of errors on a single visualization. Thus, VisuaLint must offer a composable visual vocabulary for displaying errors.
- **G4 Uncertain.** Rather than using precise displays, VisuaLint should reflect the fact that best practice in data visualization continues to evolve [Kos16]. Less precise design elements also help subvert a visualization's rhetorical force [KHAA16, Kos08]: an apparently objective chart is above flaw, but, by reducing a chart's sense of authority, we can create space for readers to consider its limitations, facilitating critique.

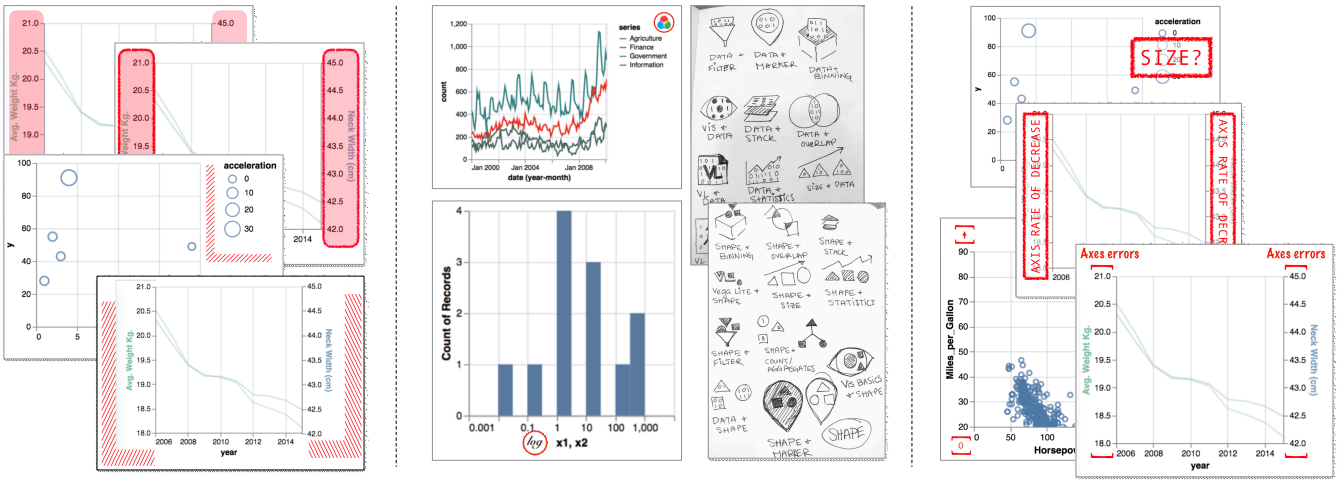

Figure 3: *Design alternatives we considered including (left to right) varieties of shading elements, iconography, and textual annotations.*

3.3. Initial Design Motifs

To ensure a broad exploration of the potential design space of visual linting techniques, we began our process with three distinct design motifs (Fig. 3) and evaluated them through informal formative interviews. Each motif went through several iterations, examples of which are included in supplemental materials.

Shading *(Fig. 3(left)).* We explored a variety of shading strategies including translucent overlays and crosshatching. This strategy seemed promising initially because it saliently conveyed "error" (**G1**) and, in the case of crosshatching, also evoked a sense of uncertainty (**G4**). However, our formative interviewees uniformly found that shading distracted from reading the visualization (**G1**). Moreover, we found the technique to have limited expressivity — how would we point out similar colors using shading, for example?

Iconography *(Fig. 3(center)).* Inspired by the "hamburger" and 3-dots "more" buttons, we considered a variety of icons to represent errors such as using incorrect scale types, perceptually ineffective color choices, and truncated axes. While these icons were unobtrusive, evaluators did not find them to be sufficiently salient (**G1**) with several participants failing to see the icons altogether. More problematically, many participants had trouble interpreting what the icons meant, a problem exacerbated by the fact that, to many participants, the icons appeared to just "float." Taken together, these results suggested the icons developed a poor mapping between the source and expression of error (**G2**).

Text Annotations *(Fig. 3(right)).* We examined several styles of textual annotation, varying the font size, placement, and verbosity of the explanation. With an appropriate styling, we found that text could be made salient (**G1**), and its description facilitated a close mapping particularly when placed alongside the erroneous element (**G2**). However, we found it difficult to concisely explain errors in situ without relying on visualization jargon (e.g. "scales", "domain", "encoding", etc.) and felt that text poorly communicated a sense of uncertainty about chart construction (**G4**).

3.4. Final Design Motif & Implementation

Our final approach is inspired by Wood et al.'s sketchy rendering technique [WII*12], a framework for presenting charts in a hand drawn style. Wood's technique is a good candidate for VisuaLint because, as they showed, sketchy visualizations increased engagement of readers and were more approachable than their traditional, austere counterparts (**G4**). They further emphasized the attention-grabbing character of sketchy graphs, stating that *"the 'disfluence' created by sketchy rendering may, in some contexts aid understanding by focusing attention of the reader"* (**G1**).

However, there are some key differences in our approach. First, for a more direct mapping to the source of the error (**G2**), we only render erroneous elements in a sketchy format rather than the entire visualization. Second, to ensure that the sketchiness does not impair reading the visualization (**G1**), sketchiness is never applied to label and title elements. Finally, to facilitate composition (**G3**) and to provide additional context, sketchy renderings of chart elements may occasionally be accompanied by sketchy label annotations (e.g., a 0 at the bottom of a sketchy y-axis to indicate the lack of a zero baseline).

We implement VisuaLint via a set of heuristics for five types of errors, shown in Figure 4. These errors include not using size encodings when negative data is involved (Fig. 4(a)); dual axis charts with differing scales (Fig. 4(b)); using perceptually ineffective colors (Fig. 4(c)); missing legends (Fig. 4(d)); and, truncated axes (Fig. 4(e)). Rather than an exhaustive demonstration, we strategically chose these errors to evaluate the efficacy of sketchy in situ annotations. In particular, to facilitate a crowdsourced study, we focus on *chart construction errors* or errors that do not require knowledge of the underlying data semantics. To demonstrate how the design language of VisuaLint can be adapted and extended, we selected errors associated with a diverse range of visual encoding choices. Moreover, these errors span a gamut of complexity to help us determine how much data literacy or visualization familiarity lay users need to understand what is being depicted, and whether VisuaLint is still helpful when errors are "obvious."

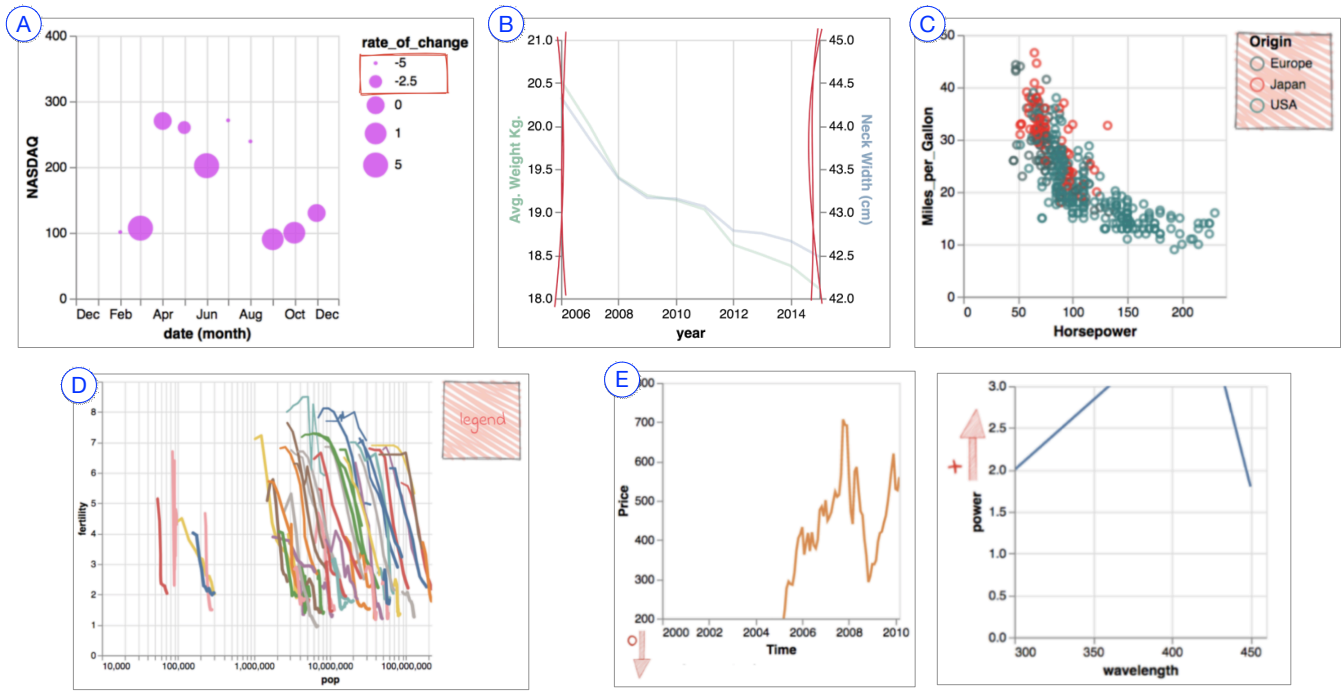

Figure 4: *We implemented VisuaLint for six types of chart construction errors: (a) inexpressive size encodings; (b) dual axis charts with differing scales; (c) perceptually ineffective color encodings; (d) missing legends; and, (e) truncated axes.*

Our heuristics operate as a layer on top of Vega-Lite visualizations [SMWH16]. We inspect parsed Vega-Lite and the rendered SVG output to identify how specific errors might manifest in Vega-Lite visualizations. For example, to identify ineffective color encodings (Fig. 4(c)), we compute a CIELAB perceptible difference value $\Delta E_{p,s}$ between color encodings on a graph; color pairs that fall below the just noticeable difference (jnd, $\Delta E_{p,s} \approx 2.0$) may not be distinguishable to all viewers, and thus are flagged. For dual axes charts (Fig. 4(b)), our heuristics identify the y-axis elements, extract the tick labels, and calculate the rate of change. Similarly, truncated axes (Fig. 4(e)), missing legends (Fig. 4(d)), and negative size encodings (Fig. 4(a)) are caught via parsed Vega Lite—when a negative value is present in a field associated with size—or rendered SVG—as is the case for missing legends and truncated axes. These heuristics are not designed to be general-purpose or tool agnostic but rather were crafted to help us evaluate the VisuaLint technique specifically. Once erroneous elements are identified, they are removed from the SVG tree and replaced with their sketchy counterparts generated through Rough.js [Shi19].

4. Evaluation

We performed a two-part, crowdsourced evaluation of VisuaLint consisting of (1) a between-subjects study measuring VisuaLint's expressiveness, comprehensiveness, and informativeness, and (2) a survey to gather qualitative feedback on VisuaLint. Using Prolific.co, we were able to quickly engage a diverse population with varied levels of data literacy. The study took an average of 35.2 minutes. Participants were recruited through the Prolific.co platform and were compensated at an average rate of $10.06/hour. Supplementing our initial informal interviews, our intent with this study was to measure whether our designs resulted in people's increased awareness of chart errors, and to simultaneously develop a baseline for lay error recognition.

All experimental materials, including stimuli and data tables, are hosted on OSF: `https://osf.io/jwbn2/`.

4.1. Between-Subjects Study Design

We used a between-subjects design, where each participant saw only one type of lint design:

- **Text**: the participant received a textual description of the error, underneath the chart.
- **Visual**: the participant received only our *in situ* VisuaLint indication of the error.
- **Visual + Text**: the participant received both the text and *in situ* error indication.

There were four phases of the experiment. A **prior** identification phase where participants saw a series of charts (each of which may or may not have a chart construction error) with no lints whatsoever and were asked to identify potential errors, an **exposure** to charts with lints exposed using the relevant lint design, a **post** identification phase, where participants repeated the prior task with a different series of (non-linted) charts, and then a final **ranking** task where participants were shown all designs on a set of sample charts and asked to rank them in order of preference while providing justifications and feedback regarding their ranking. The exposure phase additionally acted as an attention check for participants–participants were asked to click on the location of the error mes-

sage. Those that consistently did not complete this selection were not included in the final results analysis.

The charts used in the evaluation are available in our supplemental material, and were intended to represent a wide variety of common chart types with different designs and data domains. Several charts (such as Fig. 4(b) and 4(d)) replicated real-world examples that exhibited one of our selected errors, while others were specifically crafted to present one of the errors listed. Each chart consisted of one chart construction error.

4.2. Prior and Post Identification Task

Participants were exposed to one of five kinds of chart construction errors in the **prior** and **post** identification tasks (text *in italics* is the exact wording we used to surface these errors in our **Text** and **Visual+Text** conditions):

- **Color**: color was used to encode nominal data, but two categories were assigned identical or very similar colors. *The colors are too similar*.
- **Legend**: color was used to encode nominal data, but there was no legend communicating which category received which color. *There is no legend*.
- **Rate**: a dual axis encoding was used, but the rate of change in the two axes were dramatically different or otherwise non-comparable. *The axes change at different rates*.
- **Size**: size was used to encode quantitative data, but some data values were negative. *There are negative values associated with the size encodings*.
- **Axis**: the y-axis was truncated such that it did not start at zero. *the y-axis does not start at zero.*

Participants initially saw three examples each of these errors, as well as three examples of charts with none of the errors listed above, for a total of $3 \times 6 = 18$ charts in the **prior** task. A different set of 18 charts with the same error allocation were shown in the **post** task. For each chart, participants were given a binary forced-choice: *What do you think of the construction of this graph?* (1) *It's well-constructed*, (2) *There are errors in construction*. If the participant indicated the presence of errors, we then solicited a free-text response where they would describe the error.

Our primary quantitative measure for the **prior** and **post** tasks was correctness in identifying the chart error present. We assessed this through dual coding: two paper authors independently assigned a 0 (indicating that the participant's response did *not* identify the error to our satisfaction) , 1 (indicating that the participant's response *did* point to the chart error), or a *P* (indicating that the participant's response only partially pointed to the error, or that it was ambiguous whether or not the participant was correct). The coders then met to reconcile errors and generate a consensus code. There was high inter-rater reliability prior to this consensus process (Cohen's $\kappa = 0.88$, with only 5% of codes being mismatches). Responses with a consensus code of *P* were excluded from our later analyses of correctness due to their ambiguity.

4.3. Ranking Task

In addition to the four errors listed above, we presented two additional types of errors in the **ranking** task, which we excluded from our main condition as they are not detectable in the chart *per se* without an explicit lint, and so identification would not be meaningful in improving identification of errors:

- **Log Neg**: a logarithmic scale was used to place values, but there were negative values in the data (that are subsequently not plotted in the graph). *There are negative values in the data, but a logarithmic scale is being used*.
- **Range**: there are data values in the graph that exceed the domain of either the x- or y-axis of the graph and are subsequently not plotted in the graph. *Not all data is included in the axis range*.

By included these additional, data-focused errors, we hoped to surface a richer description of user preferences and needs outside of construction errors. Participants saw one set of charts for each of these seven error types, and were asked to rank four versions of each chart (one with no expressed lint, and then three with each of our proposed text, visual, or text+visual lint designs). Each set's chart presentation was varied across participants to avoid a biasing effect in ranking. We performed light, informal open-coding on ranking explanation to uncover the most relevant themes, and then grouped these themes based on commonalities.

4.4. Hypotheses

We believed that, as participants in our pool were unlikely to be familiar with all of the chart errors presented and that both visual and textual lints (or their conjunction) would educate our participants, **correctness in identifying chart errors would increase after exposure to chart lints**, regardless of lint design.

We also believed that combining both textual and visual lints would be beneficial. VisuaLint indicates the *presence* and *location* of errors, while text lints directly state *what* the error is. Therefore, we hypothesized that **the increase in correctness in identifying chart errors would be highest in the visual+text condition**.

Once primed, we believed that certain flaws would be almost impossible to miss. Indistinguishable colors, for example, are immediate points of confusion. Similarly, the absence of a legend ensures ambiguity for some facet of the graph. In contrast, using a size encoding with negative data creates confusion but does not stand out in the way a "missing" color or legend might. Participants may also not be as familiar with size encodings or dual axes compared to more common charts, and thus might be less cognizant of errors in their construction. As such, we hypothesized that **the correctness in identifying chart errors would not be uniform across all error types**.

We did not have strong hypotheses about our ranking data, beyond the rankings generally supporting our hypotheses above. That is, **participants would generally rank our linting interventions as preferable to an absence of lints** (in support of our first hypothesis) and **participants would not have uniform preferences in linting designs across errors** (in support of our third hypothesis, and under the assumption that "obvious" chart errors may require less ostentatious or redundant linting designs).

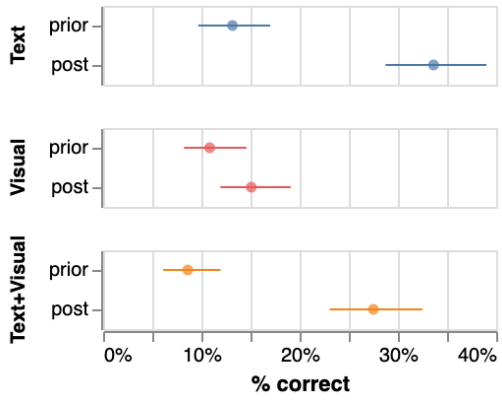

Figure 5: *Participants' prior and post accuracy (with 95% bootstrapped confidence intervals) across the three lint conditions.*

4.5. Participants

We collected data from 62 participants using Prolific. Respondents reported their gender as follows: 42% percent as female, 56% percent as male, and 2% as other or non-binary. Participants self-reported age ranges between 18-74, with 48% (n=29) of participants in the 18-24 age range, 29% as 25-34, 13% as 35-44, 5% as 45-54, 3% as 55-64, and 2% as 65-74. The most commonly reported highest completed level of education was a Bachelor's degree (37%, n=23), followed by some college credit (23%), high school graduate (21%), Master's degree (10%), Associate's Degree (4%) and trade/technical/vocational training (3%), while some high school, no diploma had the lowest response (2%, n=1).

Participants reported being "very uncomfortable" (5%, n=3), "somewhat uncomfortable" (18%, n=11), "neither uncomfortable nor comfortable" (15%, n=9), "somewhat comfortable" (47%, n=29), and "very comfortable" (16%, n=10) interpreting charts and working with data. Current occupations were very diverse, ranging from Exercise Riders (apparently a person that rides horses to keep them fit), Home Makers, Dental Technicians Analysts, Designers, Managers, to PhD students and more.

5. Results

After coding responses, we computed accuracies across each condition. We found improvement across all three intervention methods as seen in Figure 5. Overall, prior to intervention, participants correctly identified 11.15% of errors (95% CI: 0.09-0.13). Participants in the text group initially performed slightly better than their counterparts (13.16 % correct, 95% CI: 0.10-0.17), while VisuaLint + text identified a lower 8.61% % (95% CI: 0.06-0.12).

Between pre- and post-intervention conditions, there was a positive shift: as shown in Figure 5, we observed a marked increase in participants' percentage of recognized errors for each intervention. Both VisuaLint + text and text intervention participants doubled their accuracy, while just VisuaLint improved by a factor of roughly 1.5. This supports our first hypothesis, which proposed that any of our error surfacing methods would positively impact recognition.

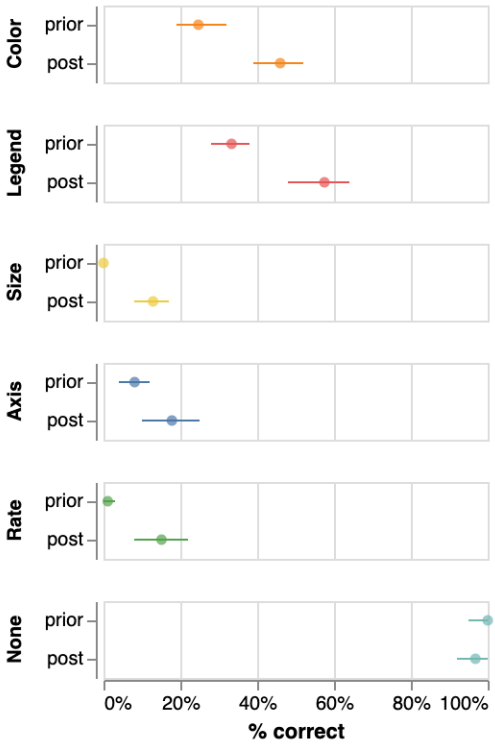

Figure 6: *Participants' prior and post accuracy (with 95% bootstrapped confidence intervals) across each error condition.*

Further, both text and VisuaLint + text conditions saw a more than doubling of errors caught.

Prior to intervention, those self-reporting as "very comfortable" interpreting graphs performed approximately the same as those that felt very uncomfortable (10.0% and 9.25 % respectively). This was not expected, but may indicate either inattention—a lack of close reading—over-confidence, or a shared, across-participant lack of awareness for key best practices in visualization. We suspect the last to be the case, as all groups clearly improved post-condition. Interestingly, accuracy rates for these "very comfortable" individuals *post-intervention* were higher than "very uncomfortable" groups. While these numbers are still numerically close (23.33 and 18.51 %), it does suggest that those familiar with visualizations and data work may require less scaffolding to inform critical perceptions of charts.

5.1. Correctness Across Designs

Our choice to not prime users with explanations of the visual signifiers of our lints was intentional; we intended to unearth unfacilitated expressiveness. From this perspective, VisuaLint did well–people understood the designs enough to internalize construction errors and improve their recognition. Our unobtrusive but salient visuals aligned with our goals, yet by prioritizing this balance we removed some quality of descriptiveness. It is therefore understandable that the VisuaLint condition did not improve participants' recognition as well as the text or VisuaLint + text conditions. However, contrary to our second hypothesis **we did not observe a con-**

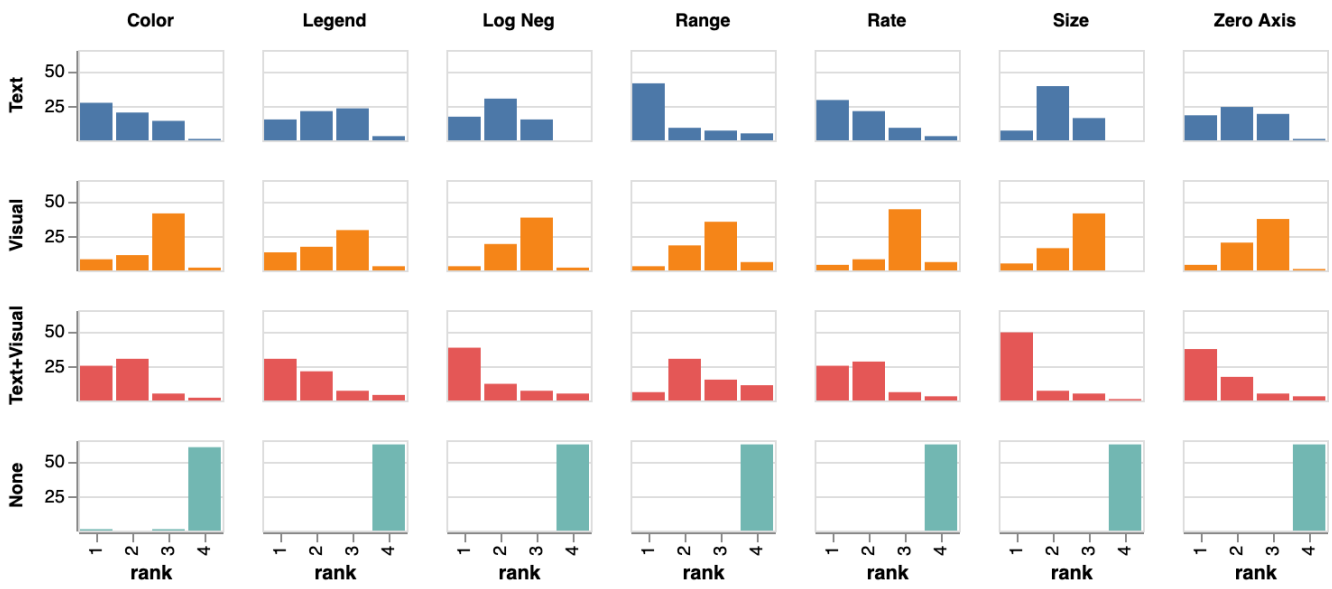

Figure 7: *Rankings of participants' preferred linting style across error types (1 is highest, 4 is lowest).*

sistent benefit for visual lints over text lints, for either VisuaLint *per se*, or VisuaLint supplemented with a textual lint.

When examining free-form error identification responses, we noticed a surprising and frequent pattern: groups that were exposed to the text only condition often perfectly copied its wording when identifying construction errors. While text is informative, it does not indicate the location of the error–less experienced participants might read but not comprehend or be able to re-apply information. This replication of structure may point to a lack of understanding of the holistic meaning of an error.

As the VisuaLint condition included no text, there was nothing to copy: participants must closely inspect a chart to recognize why the signifier appeared, thus discovering patterns of similarities *for themselves* through the different intervention examples. Some of these recognized patterns were incomplete–or incompletely described–such as in the case of these negative size encoding descriptions: "There are minus legends", or "There are negative numbers". Others showed full understanding of the indicated error, pointing out "the circle sizes are counter-intuitive to the data they are representing", and that there were "circle sizes representing negative values". This was further emphasized in the ranking task. As one person put it, "reading the text made it instantly clear what the problem was, where as the visual alone made me have [to] think further!"

5.2. Correctness Across Errors

Supporting our third hypothesis of non-uniformity in accuracy, there was a relationship between the amount of improvement and the type of error; while all error conditions showed improvement, participants responded to Color and Legend particularly well, exhibiting that**ease of error recognition is not consistent across all error types**, seen in Figure 6. During the prior phase, participants performed quite poorly across all errors: participants successfully recognized 24.7% of color errors (95% confidence interval: 0.19-0.32), 33.33% of legend errors (95% confidence interval: 0.28-0.38), 0% of size encoding errors, 8.1% of truncated axis errors (95% confidence interval: 0.04-0.12), 1.1% of rate errors (95% confidence interval: 0.00-0.03), and correctly labeled 100% of "well-constructed" graphs–those absent of errors–as such. Post-intervention, participants showed improvement in recognition across all errors, correctly identifying 46.0% of color errors (95% confidence interval: 0.39-0.52), 57.5% of legend errors (95% confidence interval: 0.48-0.64), 12.9% of size encoding errors (95% confidence interval: 0.08-0.17), 17.8% of truncated axis labels (95% confidence interval: 0.10-0.25), 15.1% of rate errors (95% confidence interval: 0.09-0.22), and incorrectly labeled 3.2 % of "well-constructed" graphs, correctly labeling 96.8% (95% confidence interval: 0.92-1.0). The *obvious* nature of color and legend errors—those that we posited to be more visually salient—was reflected in both prior and post phases. Both error classes had the highest rates of recognition in both conditions: overall accuracy for color increased from 24.73 % to 45.16 % post intervention, while post-intervention recognition of legend errors improved to 57.53%.

Despite color and legend errors being apparently *obvious*, the low rates of recognition found in our study *even with intervention* indicated a surprising disparity of data or design literacy in our user group, a disparity further exacerbated by the majority of respondents (75%) having stated their highest education level achieved as *at least* some education beyond high school.

5.3. Ranked Preferences

Figure 7 shows these results in detail: in every case, a lint of any sort was overwhelmingly preferred over a "lint-less" chart. Across

our seven error conditions, the visual+text lint was the modal first-ranked choice for four; for color, range, and dual axis rate errors, the text-only lint was preferred.

In the responses ranking, we found that there was a difference in how people perceived the usefulness of an error expression. For errors that related to information users were likely familiar with–the absence of a legend and similarity of color–participants consistently rated visual highest. One participant stated "since the legend is missing, the visual is clear as to what the problem is, and stand out more than the text"

But for more complex errors, as one respondent pointed out, "text is useful due the complexity of the error." This does not mean that text alone was sufficient. In general, users preferred VisuaLint with text: "The text gives context for the visual, the visual by itself doesn't give a reason for why it's there. On the other hand, the text by itself is easy to miss." Reinforcing this, in the context of a truncated axis, "Text and visual give a complete showing of where the error is. The placement of the error and the name of the error. Visual on its own is fine, text on its own gives less information."

6. Discussion and Future Work

General low rates of error detection would seem to point to a broad lack of data and visualization literacy. With the exception of post-intervention legend mistakes, participants consistently scored below 50% accuracy. Even participants who self-described as "very comfortable" did not catch errors at a frequency we would expect. And while different methods of error expression may augment this awareness–both in the short-term, as seen from the ranking responses, and in the longer-term internalization of heuristics, as shown in the post-condition identification task–it cannot replace the deeper engagement that comes from courses, books, and other more verbose materials.

In order for signifiers to be effective, users must understand what they reference. For populations less familiar with chart best practices, there may be great benefit in an educational scaffolding approach [She05]. This scaffolding follows a format in which users are initially given a myriad explanations, a number which, following their development of expertise, is rapidly reduced. This suggests that users *need* verbose explanations for new material–as text and VisuaLint + text interventions provide–and reemphasizes an earlier point: that people lack an intuition for visualizations. As users further develop their expertise, this demand dissipates (as seen in both color and legend errors). By scaffolding VisuaLint's descriptiveness, we might allow users autonomy of choice. One method, borrowing again from spell check, may include an interaction with VisuaLint that furnishes users with verbose explanations and provides automated chart corrections. This references the functionality of later spell check tools, which not only caught and surfaced errors, but provided recommendations and even automated adjustments.

A natural next step is integrating VisuaLint in a broader system, such as the Vega Editor or Observable. Errors of construction are common in both, yet the current method of surfacing includes ambiguous messages conflating code errors with construction errors. These messages are difficult to parse, particularly for novice readers. For visualization experts, who do not need verbose explanations to understand construction errors, VisuaLint offers a unique solution in it's unobtrusive, in situ approach to efficacy checks.

Parallel to our interest in subverting the rhetorical force of visualizations is that of trust. Do the efficacy checks provided by VisuaLint impact readers' trust in a visualization? This is particularly relevant in context of data journalism, where narratives in visualizations inform a broad population.

In general, exposing conventions of visualization practices may highlight implicit assumptions of visualization authors, inform their readers of said practices, and generally improve the quality of visualizations across many domains. VisuaLint offers a glimpse into the future user experience of visualization linters—by focusing on authors *and* readers, we facilitate a communion between both groups. Ultimately, we hope future work builds on three elements of this paper: 1) developing informed critique of visualizations, 2) apprising users of visualization errors, and 3) further exploring visualization literacy in a broad audience.

7. Acknowledgements

We thank our anonymous reviewers as well as Nava Haghighi, Jonathan Zong, Crystal Lee, Alan Lundgard, and other members of the MIT Visualization Group for all their thoughtful feedback. We would further like to acknowledge the online community of data visualization experts, journalists, and authors for their continued work on catching poorly constructed visualizations and improving data communication. This work was supported by NSF Award 1900991.

References

[BBC*01] Blackwell A. F., Britton C., Cox A., Green T. R., Gurr C., Kadoda G., Kutar M., Loomes M., Nehaniv C. L., Petre M., et al.: Cognitive dimensions of notations: Design tools for cognitive technology. In *International Conference on Cognitive Technology* (2001), Springer, pp. 325–341. 2, 3

[BBZ18] Barowy D. W., Berger E. D., Zorn B.: Excelint: Automatically finding spreadsheet formula errors. *Proceedings of ACM Programming Languages 2*, OOPSLA (Oct. 2018), 148:1–148:26. doi:10.1145/3276518. 2

[BDSK*17] Binnig C., De Stefani L., Kraska T., Upfal E., Zgraggen E., Zhao Z.: Toward sustainable insights, or why polygamy is bad for you. In *CIDR 8th Biennial Conference on Innovative Data Systems Research* (2017). 2

[BE15] Bresciani S., Eppler M. J.: The pitfalls of visual representations: A review and classification of common errors made while designing and interpreting visualizations. *Sage Open 5*, 4 (2015). doi:10.1177/2158244015611451. 2

[BGB14] Barowy D. W., Gochev D., Berger E. D.: Checkcell: Data debugging for spreadsheets. *Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications OOPSLA 49*, 10 (2014), 507–523. doi:10.1145/2660193.2660207. 2

[DAREA*18] Diehl A., Abdul-Rahman A., El-Assady M., Bach B., Keim D., Chen M.: Visguides: A forum for discussing visualization guidelines. In *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers* (2018), pp. 61–65. 3

[GBK17] GUO Y., BINNIG C., KRASKA T.: What you see is not what you get!: Detecting simpson's paradoxes during data exploration. In *ACM SIGMOD Workshop on Human-In-the-Loop Data Analytics (HILDA)* (2017), ACM, pp. 2:1–2:5. doi:10.1145/3077257.3077266. 2

[HSH18] HOFFSWELL J., SATYANARAYAN A., HEER J.: Augmenting code with in situ visualizations to aid program understanding. In *ACM Human Factors in Computing Systems (CHI)* (2018). URL: http://vis.csail.mit.edu/pubs/insitu-vis-debugging. 3

[HST17] HYNES N., SCULLEY D., TERRY M.: The data linter: Lightweight, automated sanity checking for ml data sets. In *NIPS: Workshop on Systems for ML and Open Source Software* (2017). 3

[KHAA16] KENNEDY H., HILL R. L., AIELLO G., ALLEN W.: The work that visualisation conventions do. *Information, Communication & Society 19*, 6 (2016), 715–735. 2, 3

[Kos08] KOSTELNICK C.: The visual rhetoric of data displays: The conundrum of clarity. *IEEE Transactions on Professional Communication 51*, 1 (2008), 116–130. doi:10.1109/TPC.2007.914869. 2, 3

[Kos16] KOSARA R.: An empire built on sand: Reexamining what we think we know about visualization. In *Proceedings of the sixth workshop on beyond time and errors on novel evaluation methods for visualization* (2016), ACM, pp. 162–168. 2, 3

[KS08] KIRBY R. M., SILVA C. T.: The need for verifiable visualization. *IEEE Computer Graphics and Applications 28*, 5 (2008), 78–83. doi:10.1109/MCG.2008.103. 2

[LDH*19] LEE D. J. L., DEV H., HU H., ELMELEEGY H., PARAMESWARAN A. G.: Avoiding drill-down fallacies with vispilot: Assisted exploration of data subsets. In *Proceedings of the 24th International Conference on Intelligent User Interfaces IUI* (2019), ACM, pp. 186–196. doi:10.1145/3301275.3302307. 2

[Leo19] LEO S.: Mistakes, we've drawn a few. https://medium.economist.com/mistakes-weve-drawn-a-few-8cdd8a42d368, March 2019. 1, 3

[Lou06] LOURIDAS P.: Static code analysis. *IEEE Software 23*, 4 (2006), 58–61. 3

[Mee17] MEEKS E.: viz-linting. https://github.com/emeeks/viz-linting, May 2017. 2, 3

[MK18] MCNUTT A., KINDLMANN G.: Linting for visualization: Towards a practical automated visualization guidance system. In *VisGuides: 2nd Workshop on the Creation, Curation, Critique and Conditioning of Principles and Guidelines in Visualization* (2018). 2, 3

[MKC20] MCNUTT A., KINDLMANN G., CORRELL M.: Surfacing visualization mirages. In *ACM Human Factors in Computing Systems (CHI)* (2020). to appear. 2

[MWN*19] MORITZ D., WANG C., NELSON G. L., LIN H., SMITH A. M., HOWE B., HEER J.: Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 438–448. doi:10.1109/TVCG.2018.2865240. 2, 3

[PAEE19] PECK E. M., AYUSO S. E., EL-ETR O.: Data is personal: Attitudes and perceptions of data visualization in rural Pennsylvania. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), ACM, p. 244. 3

[PRS*15] PANDEY A. V., RALL K., SATTERTHWAITE M. L., NOV O., BERTINI E.: How deceptive are deceptive visualizations?: An empirical analysis of common distortion techniques. In *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems* (2015), ACM, pp. 1469–1478. doi:10.1145/2702123.2702608. 2

[She05] SHEPARD L.: Linking formative assessment to scaffolding. *Educational Leadership* (2005). 9

[Shi19] SHIHN P.: Rough.js: Create graphics with a hand-drawn, sketchy, appearance. https://roughjs.com, December 2019. 5

[SML*18] SAKET B., MORITZ D., LIN H., DIBIA V., DEMIRALP C., HEER J.: Beyond heuristics: Learning visualization design. *arXiv preprint arXiv:1807.06641* (2018). 2

[SMWH16] SATYANARAYAN A., MORITZ D., WONGSUPHASAWAT K., HEER J.: Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2016), 341–350. doi:10.1109/TVCG.2016.2599030. 2, 3, 5

[Tuf01] TUFTE E. R.: *The visual display of quantitative information*, vol. 2. Graphics press Cheshire, CT, 2001. 3

[WII*12] WOOD J., ISENBERG P., ISENBERG T., DYKES J., BOUKHELIFA N., SLINGSBY A.: Sketchy rendering for information visualization. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2749–2758. 2, 4

[ZZZK18] ZGRAGGEN E., ZHAO Z., ZELEZNIK R., KRASKA T.: Investigating the effect of the multiple comparisons problem in visual analysis. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), ACM, p. 479. 2