

Teru Teru Bōzu: Defensive Raincloud Plots

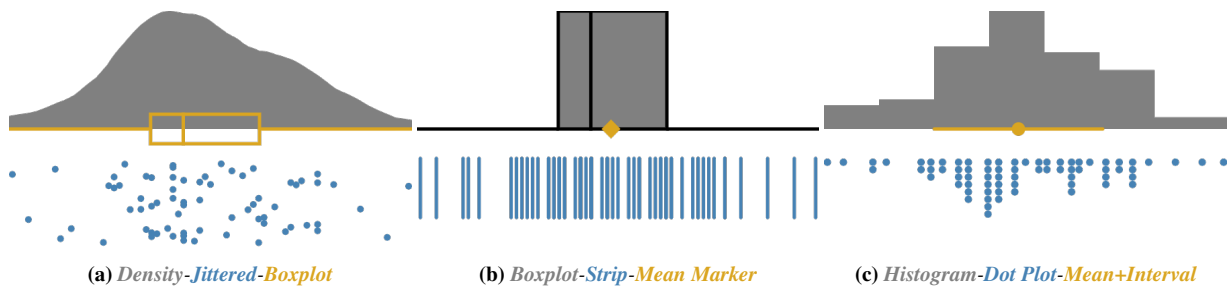
Michael Correll¹ ¹ Work completed while at Tableau Research

Figure 1: Three examples of raincloud plots showing the distribution of the yearly number of days with precipitation in Seattle, 1948-2017. A raincloud plot is made of three component visualizations: *clouds* that provide the overall shape of the distribution, *rain* that plot individual data values, and optional *lightning* for additional derived or inferential statistics.

Abstract

Univariate visualizations like histograms, rug plots, or box plots provide concise visual summaries of distributions. However, each individual visualization may fail to robustly distinguish important features of a distribution, or provide sufficient information for all of the relevant tasks involved in summarizing univariate data. One solution is to juxtapose or superimpose multiple univariate visualizations in the same chart, as in Allen et al.'s [APW*19] “raincloud plots.” In this paper I examine the design space of raincloud plots, and, through a series of simulation studies, explore designs where the component visualizations mutually “defend” against situations where important distribution features are missed or trivial features are given undue prominence. I suggest a class of “defensive” raincloud plot designs that provide good mutual coverage for surfacing distributional features of interest.

CCS Concepts

• *Human-centered computing* → *Visualization systems and tools; Visual analytics; Visualization techniques;*

1. Introduction

Prominent examples like Anscombe’s quartet [Ans73] or the Datasaurus dozen [MF17] show that concise sets of summary statistics may fail to identify distributional features of interest. Visualizations can augment these numerical summaries and provide more context or detail. Yet, simple visualizations like box plots or error bars can similarly fail to surface important distributional features [CM05, PHKD06], or, worse, be misinterpreted [CG14]. Even more complex univariate visualizations like density plots and histograms can fail to reliably surface outliers, missing data, or multimodality [CLKS19].

The deficiencies of individual classes of univariate visualizations to support the many, occasionally conflicting tasks involved in univariate distributions have led to a growing number of examples of

plots that juxtapose or superimpose *multiple* univariate charts together, where each design is intended to support different sets of tasks relating to the data distribution. The terminology for this class of design varies, for example “hybrid plots” [BDL*20, PKRJ10], “ensemble plots” [CLKS19], or “RDI plots” (standing for “Raw (data), Description and Inference”) [Phi17]. For this paper, I anchor on a specific class of such designs: **raincloud plots** [APW*19] (Figure 1). As with *teru teru bōzu*, the Japanese good luck charms for warding off rain [Sto06], these rain cloud plots ward off situations where viewers might miss critical information about a data distribution, and promise a sunny day of visual analytics based on trusted and well-understood data.

Raincloud plots juxtapose a “cloud” that provides a summary of the overall shape or extent of the distribution (such as a density plot or violin plot) with “rain” that plots individual data points (such as

a jittered dot plot or strip plot). The clouds and rain can be augmented with additional marks indicating medians, normal modes, or inferential statistics [PKRJ10]. I was unable to identify an appropriate extent term for these design elements and so, to extend the weather metaphor, I refer to them as “**lightning**.” The cloud, rain, and lightning, together, present multiple facets of the distribution, at multiple scales, and in the same plot: as per Allen et al. [APW*19], “the reader has all information needed to assess the data, its distribution, and the appropriateness of any reported statistical tests” all “in an appealing and flexible format with minimal redundancy.”

Raincloud plots are still relatively uncommon, earning an entry in Lambrechts’ collection of “**xenographics**”—rare, strange, or unfamiliar forms of visualizations. Even within their niche, there appears to be some disagreement about what elements they must or should contain, or even their general utility. In this paper, I categorize and assess the individual design components of the raincloud plot from the perspective of what Allen et al. [APW*19] call “robustness” and I call **defensiveness**: the ability of raincloud plots to consistently and reliably surface distributional features of interest even in the face of potentially adversarial settings. This defensive analysis, in turn, draws on prior work that employs simulation to assess the robustness or reliability of visualization designs [CC21, McN21, MKC20] from the perspective of **algebraic visualization design** [KS14] (AVD). That is, two raincloud plots should be visually distinct only to the degree that they represent commensurately distinct underlying distributions and, conversely, two raincloud plots should be visually similar if and only if the underlying distributions encoded by both are commensurately similar.

In this paper, I identify algebraic weaknesses in the underlying univariate visualizations that make up raincloud plots, and how these weaknesses can impact the ability of the resulting plots to act defensively to reveal distributional anomalies. As the result of this analysis, I suggest a set of best practices for raincloud plot design, including avoiding jitter dot plots in favor of more algebraically consistent visualizations of raw values such as strip plots, and towards more compact designs that reinforce the *mutually defensive* aspects of the component plots where the designer accepts some potential deficiencies in individual raincloud components (such as overplotting in dot plots, or oversmoothing in density plots) trusting the other components to “pick up the slack” and communicate what might be missed when looking at one component at a time.

2. Raincloud Plots

There are many potential reasons for visualizing a distribution (see Blumenschein et al. [BDL*20] for a summary of both individual distributional tasks as well as comparative distributional tasks). Beyond direct inspection or comparison, a viewer might wish to “sanity check” [CLKS19] a distribution to assess the quality of the data or verify that the preconditions for inferential statistics or aggregation have been met. Common summaries of visualizations like box plots or confidence intervals provide *some* of the information relevant to such task, but not *all* of it. These simple plots also can produce ambiguities. A boxplot, for instance, can obscure the nature of an underlying distribution, reducing unimodal, bimodal, and skewed distributions to similar or even identical visualizations [CM05, MF17]. Plotting the raw data directly (with one mark

per data value), such as with a dotplot [Wil99], wheat plot [Few17], or strip plot, supports different sets of tasks, but comes with its own drawbacks. For instance, higher-level summary statistics are no longer directly encoded but must be estimated through ensemble visual processes [SHGF16], which may or may not be sufficiently accurate for statistical purposes, as these estimates are likely driven by “perceptual proxies” [YHF19] rather than explicit statistical calculation. Additionally, as the number of data points becomes large, overplotting or other issues of scale further complicate the process of extracting relevant statistical information and also impact the sheer legibility of the chart.

If high-level visualizations of summary statistics are insufficient to reveal the interior structure of distributions, but low-level visualizations of raw data values are insufficient to reveal higher-level moments or scale to large datasets, a natural solution is to combine the two into a single, mutually-supportive visualization. There have been many proposed solutions in this space. For instance, the canonical violin plot [HN98] combines a mirrored density trace with an interior box plot, whereas the bean plot [K*08] replaces the box plot with an interior strip plot. Of special note are two “kitchen sink” ensemble plots: the summary box plot [PKRJ10] where the mirrored density trace is augmented with redundant density information as well as glyphs for not just means and medians but also higher moments such as skew and kurtosis, and the v-plot [BDL*20] generalization for custom, dynamic creation of arbitrary ensemble plots for the comparison of distributions (for instance, histograms superimposed on density traces, or violin plots juxtaposed with “dynamite plots” of bar charts with error bars).

Of recent prominence in the space of ensemble plots is the raincloud plot, so dubbed by Allen et al. [APW*19]. Extending the weather metaphor in the paper, I observe three components in the raincloud plot design, and use a consistent color scheme in the figures and text in the remainder of this paper to differentiate them:

- **Clouds**: high-level summaries of the distribution at a level above the raw data. For instance, density plots (Figure 1a), boxplots (Figure 1b), or histograms (Figure 1c).
- **Rain**: low-level plots of the individual data values. For instance, jittered dot plots (Figure 1a), strip plots (Figure 1b), or Wilkinson-style dot plots [Wil99] (Figure 1c).
- **Lightning**: glyphs or other visualizations of derived statistics from the distribution. For instance, confidence intervals (Figure 1c), mean values (Figure 1b), or even glyphs encoding higher moments like skew or kurtosis [PKRJ10]. Note that this term is my own, and there is some ambiguity about whether, e.g., a box plot fulfills either a “cloud” or “lightning” role, which is perhaps more of a function of visual design than an inherent distinction; in Figure 1a, for instance, the boxplot is in the “lightning” role, whereas it takes a “cloud” role in Figure 1b.

While Allen et al. [APW*19] propose several potential combinations of “rain”, “cloud”, and “lightning” in their paper, a dominant design (and the default of the associated {raincloudplots} R package) is a raincloud where the cloud is a **Density** plot (referred to in the paper as a “split-half violin”), the rain is a **Jittered** dot plot (that is, each data value is plotted as in a scatterplot but where the y-position of the dot is drawn uniformly at random), and the lightning is a **Boxplot**: Figure 1a is an example of this canonical design. As a

back-of-the-envelope assessment of the impact of these design defaults, I conducted a Google Image Search on the term “raincloud plot” and collected the first 50 relevant examples (corpus and coding available in supplement). A hand-coded classification of these examples found that **Density** plots were the overwhelming choice for clouds (48/50 examples): an exception was the use of a **Boxplot** in the cloud role. **Jittered** dot plots were similarly common as a choice of rain design (40/50), although there were solo occurrences of **Beeswarm** plots, Wilkinson-style **Dot Plots** [Wil99], and even **Heatmaps** encoding density information. Lightning had the most variability, with **Boxplots** being the norm (38/50), with the other dominant design being a **Marker** indicating a central tendency (such as the mean), either with (5/50) or without (2/50) lines for **Intervals** such as standard error or 95% confidence intervals (although none of these intervals were labelled in the figure per se; their meaning was often relegated to captions, code snippets, or often unspecified entirely, a common practice [CG14] that can lead to confusion). A minority (4/50) of examples had no lightning at all, but relied on the clouds and rain alone to communicate distributional properties.

Despite the dominance of the “**Density-Jittered-Boxplot**” raincloud, I take both the variety of examples in the original Allen et al. [APW*19] work and the proliferation of unique examples “in the wild” as evidence that there is both mutability in the concept of a raincloud and opportunity to suggest alternative designs. Given the large number of designs for visualizing distributions and the combinatorial explosion introduced by the unification of these designs, the resulting space of potential rainclouds is quite large, and could conceivably be stretched to include a number of prior ensemble plots not otherwise thought of as rainclouds (for instance, a bean plot could be characterized as a “**Mirrored Density-Strip Plot-Mean Marker**” raincloud plot). Rather than exhaustively explore this space (if such a list is even possible given the lassitude in definitions around each component: can textual annotations be a form of lightning, for instance?), in the following sections I instead present a “visualization zoo” [HBO10] of both dominant designs as well as alternative designs of potential empirical interest or prominence in the visualization literature.

2.1. Designing Clouds

As the purpose of a cloud is to provide an overview of the shape of the distribution without plotting individual points, relevant information for cloud designs are kernel density estimates (KDE), bins, or quantile information, visualized through common techniques such as **Density** plots, **Histograms**, and **Boxplots**, respectively. Figure 2 shows a few designs I either observed in the dataset, or suggest as alternatives.

For KDE-driven density plots, prior ensemble designs like violin plots and bean plots often *mirror* the density information. Per Allen et al. [APW*19], “violin plots mirror the data density in a totally uninteresting/uninformative way, simply repeating the same exact information for the sake of visual aesthetic.” In addition to redundancy, I also note that mirroring both removes the y-axis baseline for direct comparisons of density and creates negative space between distributions that may or may not reflect genuine distributional features for comparison. Empirically, however, I note that

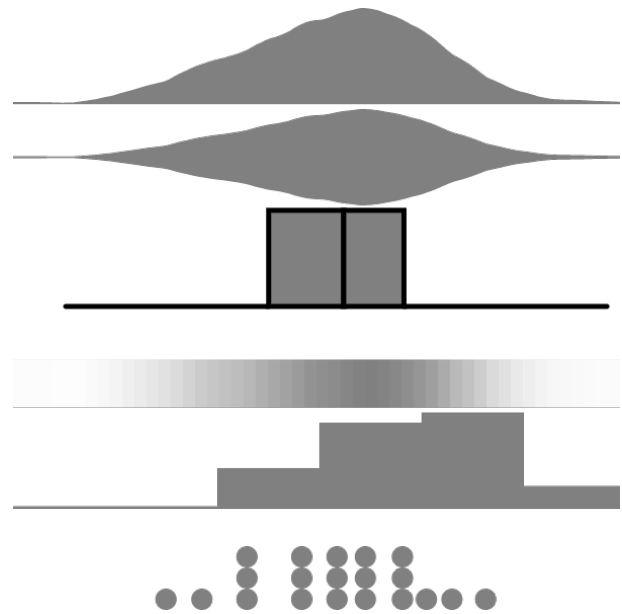


Figure 2: Potential designs for clouds on a sample dataset of 100 values drawn from a Gaussian. From top to bottom: *Density plot, mirrored density or Violin plot, “split-half” Boxplot, Heatmap, Histogram, and Quantile Dot Plot as in Kay et al. [KKHM16].*

Ibrekk & Morgan [IM87] find this mirroring to not have a significant deleterious effect on performance, and suggest potential benefits in terms of directing attention away from values and into overall shape and area: “our intent in this [mirrored] display was to try to focus subjects’ attention on the *area* between curves.” Thus, while I argue for the use of non-mirrored (“split violin”) **Density** plots, and this mirroring is largely inconsequential to the algebraic analysis in this paper, I await further empiricism on this matter.

I note that line and area charts (mirrored or otherwise) are not the only way of communicating density information. I call out two specific designs for having records of potential empirical benefits at distributional tasks. The first are quantile **Dot Plots**, as introduced in Kay et al. [KKHM16] and examined in follow-on work [FWM*18, KKH21]. Quantile dot plots are calculated via a selection of n quantile values and plotted as stacked dots as in a Wilkinson dot plot [Wil99]. The resulting plot approximates the overall shape of the distribution but, unlike in a dot plot of raw values, the number of points shown are capped, affording estimates that rely on counting (for instance, once could count the number of dots below some threshold, and make a frequency-based judgment such as “in 5 out of n cases, the value is less than the threshold”). While I find quantile dot plots promising for showing the overall shape of a distribution in a bounded-complexity way, I note that, in a raincloud where they are juxtaposed with with a rain component that *does* show raw values, there is potential ambiguity. For instance, a **Dot Plot-Dot Plot** raincloud would likely be confusing to interpret, as there would be two very similar cloud and rain components with very different interpretations (see Figure 9 for an example). The second alternative design for distribu-

tion shapes are **Heatmaps** of the KDE. Work by Albers Szafir et al. [ACG14, SHGF16] suggests that, while color is less precise as an encoding channel for density information than position or area as used in other designs, they do afford higher precision in extracting *aggregate* statistical information such as regions of high average value or variance. In rainclouds, where the cloud is supplemented by rain and lightning for the purpose of extracting more detailed information, a **Heatmap** could function as a compact and useful summary. Two-tone [SMY*05] or horizon charts [HKA09] represent a potential compromise between traditional density charts and heatmaps, allowing the use of color to draw the eye to salient peaks or valleys of the KDE while maintaining accuracy at extracting individual density values.

A concern shared amongst many of these rain designs is selecting an appropriate kernel and kernel bandwidth (in the case of KDE-driven visualizations) or binning scheme (in the case of histograms). This problem is to some extent a bias-variance trade-off: if the kernel is too large (or the bins too wide), then fine details of the distribution (like areas of missing data) are lost, but if the kernel is too small (or the bins too narrow), the overall shape of the distribution is lost and spurious visual features like spikes and gaps appear that are a result of overfitting and sampling error rather than genuine modes or gaps in the data [CLKS19]. Various rules of thumb for setting these parameters exist (e.g. Sturges' rule [Sco09], the Freedman-Diaconis rule [FD81], Silverman's rule [She04], etc.) based on properties of the underlying data. See Correll et al. [CLKS19] for discussion of the algebraic robustness of these parameter settings.

2.2. Designing Rain

The rain component of a raincloud plot is meant to plot raw data values directly (see Figure 3 for examples). A primary design consideration is therefore the *scalability* and *legibility* of such plots as the number of data values to be plotted increases. For instance, a box plot representing a hundred values can be drawn with the same number of lines as a box plot representing a million, but the same is not true of a rug plot or strip chart. *Overplotting* is perhaps the most well-studied of these scalability issues. I discuss two solutions to overplotting in univariate data that retain the desired outcome of plotting every data value [Few08]: jittering, and adjusting opacity.

Jittering alters the location of marks by introducing positional offsets. A very simple approach to jittering (and the one encountered most frequently in my dataset of example rainclouds) is to, in the case of marks distributed along an x-axis, to introduce a (meaningless) y-axis and place points randomly within a particular range. In addition to being potentially confusing (as the y-axis does not encode any actual data), this jittering is very disruptive. Few [Few17] calls jittering “a carpet bombing that alters the entire landscape (i.e., all of the values) rather than as a surgical strike that targets only those values that are subject to over-plotting and only to the degree that is necessary to resolve the problem.” Beeswarm plots [Ek116] are a jittering approach that is less drastic, retaining the secondary axis, but packing marks as closely together as possible. The resulting envelope of the “swarm” of points resembles a violin plot. Depending on the data density and the packing algorithm used (the beeswarms in this work are generated via d3's [BOH11]

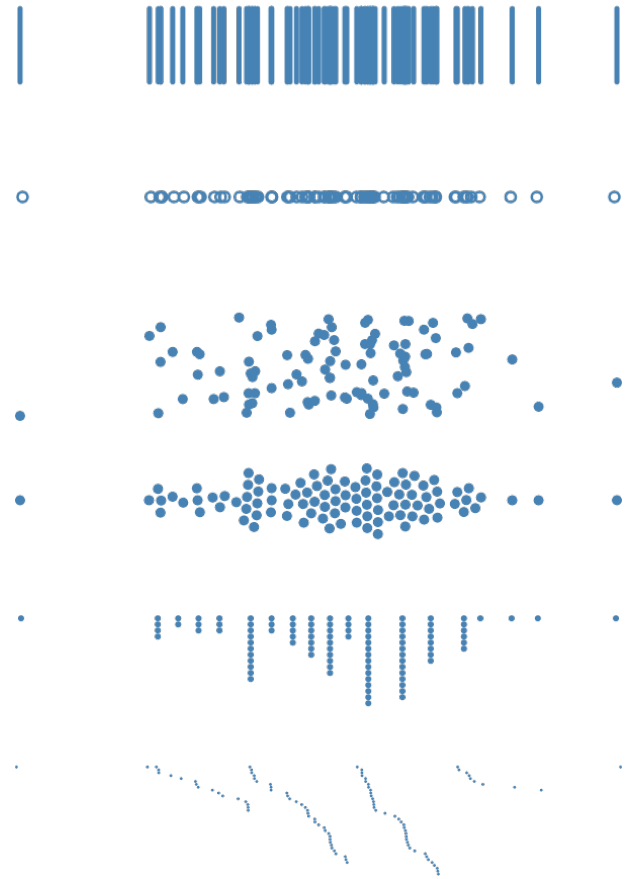


Figure 3: Potential designs for rain on a sample dataset of 100 values drawn from a Gaussian. From top to bottom: *Strip Plot charts*, *Dot Plot*, *Jittered Dot Plot*, *Beeswarm*, *Wilkinson Dot Plot* [Wil99], and *Few Wheat Plot* [Few17]

force-directed simulation functions, for instance), individual marks may end up some distance from their “actual” x-axis location. Other jittering approaches make different tradeoffs to respect positional fidelity. For instance, Wilkinson dot plots [Wil99] drop marks in columns. If a mark would be dropped in a column where it would overplot a neighbor, it is instead stacked on top of this mark, and the entire column shifted to represent the mean value of its contents (I note as an aside that the resulting dot plots in the Wilkinson paper are displayed in raincloud-like ensemble plots of dots, histograms, density plots, and strip plots all together). Lastly, Few proposes wheat plots [Few17], a sort of hybrid of a dot plot and histogram, where n discrete bins are created, and all the components of each bin are sorted, plotted at their exact x-position, but with a y-position dependent on the index within the bin (the first mark of height h in a particular bin is placed at $y = 0$, the next at $y = h$, and the i th at $y = hi$, for instance), creating the “appearance of rows of wheat bending in the wind” [Few17]. In all of beeswarms, Wilkinson dot plots, and wheat plots, the y-axis now (implicitly or otherwise) communicates density information and distributional shape.

However, it does so at the cost of compactness: where, in order to fit marks into a given region size chart and so preserve the guarantee of no overplotting, the mark size must be constrained or reduced, eventually producing legibility issues as the size of the data becomes larger or the plot size smaller. Figure 3's **Wheat Plot** illustrates this issue: with only a small number of bins, each mark has a radius of ≈ 2 pixels in order to accommodate the height of the largest "stalk" of wheat; compare to the 5 pixel radius of the **Jittered** dot plot (a more than 6x increase in area).

While positioning can entirely eliminate overplotting, it is also possible to ameliorate its effects by altering how individual marks are rendered. This can be done by reducing the size of the marks, or plotting with an empty rather than filled mark. The latter approach is reflected in the use of an empty circle as the default mark for dot plots and scatterplots in systems with recommendations or smart defaults like VegaLite [SMWH17], Voyager [WQM*17], and Tableau ShowMe [MHS07]. Reducing the opacity of marks is another approach. Again, as with histogram binning or KDE, there is a tradeoff when altering mark opacity. If the opacity is too low, then individual marks are lost to the background and only the mode(s) are salient. If the opacity is too high, then overplotting means it is difficult or impossible to compare the relative density of dense regions. While there exist automated data-driven methods for setting opacity [MAF15, MPOW17], opacity is often set to a constant by default, and must be interactively adjusted by the chart designer [MAF15].

One last concern with rain that discourages its use as the only visualization to encode data for which distributional properties are relevant is the difficulty with which the statistics of the overall distribution can be visually estimated from the individual values alone. There are several biases or "perceptual proxies" [OYK*21, YHF19] potentially present when estimating, for instance, the mean of a group of points. Estimates of the overall shape of the distribution are also potentially biased; for instance, Newburger et al. [NCE22] report a (coincidentally named for the purposes of this paper) "umbrella effect" where estimates of distributional shape are adjusted to "cover" all observed values, overestimating the variability and estimated density of the distribution in the tails.

2.3. Designing Lightning

"Lightning" is my grab-bag term for additional marks, annotations, or supplemental charts in a raincloud plot, specifically those meant to directly communicate summary or inferential statistics that are not encoded directly in the cloud or rain components. Figure 4 presents a set of examples, again based on both observations of raincloud plots as well as suggestions of potentially useful representations from the literature. These statistics can be both descriptive (for instance, means and medians, quartiles or bounds) or inferential (such as confidence intervals, predictive intervals, or indications of significant difference from some threshold). Lightning is an opportunity for the designer to directly and saliently encode information that would be difficult or imprecise to extract visually from the other components.

Despite the comparative simplicity in pre-computing and then directly plotting distributional features, there are still important

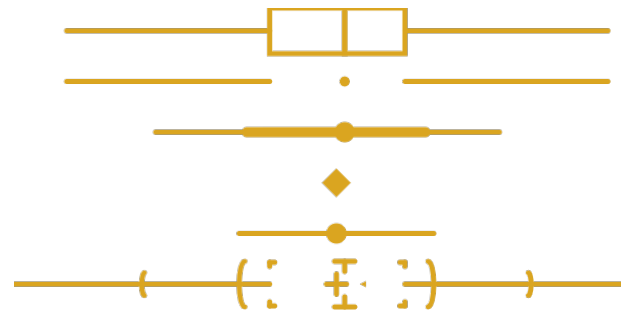


Figure 4: Potential designs for lightning on a sample dataset of 100 values drawn from a Gaussian. From top to bottom: **Boxplot**, **Midgap** plot, **QInterval** showing median, 66% quantile intervals (thick line), 95% quantile intervals (thin line), as in the [ggdist] R package [Kay21], **Mean Marker**, **Mean+Interval** (in this case, one standard deviation), and an adapted form of the Potter et al. [PKRJ10] **Moment Plot** showing mean (as a + -shaped marker), median (as "T"-shaped markers), quartiles (as an "abbreviated" box), skew (as a triangle), one and two standard deviations (as larger and smaller "("-shaped markers), and range (as lines).

considerations for designers and a multiplicity of final designs. For instance, there are many variations of the canonical **Boxplot** [PHKD06]: the "whiskers" can encode the full range of the data, or a scalar multiple of the interquartile range with points outside rendered (or not) with individual glyphs. **Boxplots** can be further notched to provide guidance on the potential significance of differences between medians, or have variable width to denote sample size, or have continuously varying width driven by a KDE, similar to a violin plot (as in vase plots [Ben88]). Lastly, the (relatively large) visual area of the box can introduce a bias in which the widths of the whiskers are underestimated: **Midgap** [SB91] plots are therefore an alternative that still convey identical quartile information while eschewing the central box. A particular variation of note are **Moment Plots** as described by Potter et al. [PKRJ10], where box plots are augmented with glyphs representing not just mean and standard deviation but also higher moments such as skew, kurtosis, and tailedness, providing detailed information about the shape and tendencies of a distribution.

I suggest three classes of potential design issues for lightning in the raincloud setting: *ambiguity*, *variability*, and *interpretability*. Lightning is often ambiguous because identical visual designs can have several meanings with different implications for how these designs should be interpreted. The whiskers of the box plot are one example [PHKD06] (encoding either range, 1.5·IQR, or the position of the nearest actual value to 1.5·IQR). **Mean+Interval** designs are also ambiguous, because the intervals can represent many different quantities [CG14] (standard deviation, standard error, z- or t-confidence intervals at various α levels, Bayesian predictive intervals, quantile intervals, etc.), all of which have different implications with respect to interpretation. Lightning is also highly variable, as observed in demonstrations such as Dragicevic's "dance of the p-values" [Dra16]: data from the same underlying sampling distribution can produce very different inferential statistics and in-

tervals: even with no underlying signal, with enough samples and enough plots, these statistics and their visualization can lead to spurious “insights” [ZZZK18]. Lastly, lightning requires varying levels of statistical familiarity to interpret; while introductory statistics classes may introduce means, medians, and quartiles, lightning visualizing higher moments or more complex derived statistics may be unfamiliar. There is also the task of going from the visual representation of a statistic to a statistical judgment or decision (for instance, using a pair of intervals to estimate the size and reliability of a difference in means between two sample populations). This sort of “inference by eye” [CF05] is non-trivial, and many viewers use “satisficing” strategies for this task that can result in biases and inaccuracies in judgments [KKH21]

3. Defensive Analysis

The goal of raincloud plots as specified in Allen et al. [APW*19] is “robustness.” I interpret this to mean a sort of mutual coverage of the weaknesses of the component parts. For instance, if overplotting in a **Strip Plot** makes it difficult or impossible to assess the relative density of a distribution, a **Density** plot provides that context. Likewise, if implicitly estimating the mean value from a **Strip Plot** using visual ensemble processes is difficult, a **Mean Marker** could directly encode that information. I use the term *defensive raincloud* to denote a raincloud in which the subcomponents are mutually supportive in this way. Determining the defensive characteristics of a raincloud, in turn, requires an analysis of the strengths and weaknesses of the subcomponents.

In keeping with prior work examining the robustness of charts [CLKS19, CC21, MKC20, McN21], I use the principles of *algebraic visualization design* [KS14] (AVD) to define potential failures of robustness. While I lay out these failures in more detail in the following sections, in general AVD failures can be thought of mismatches between changes in data and their resulting instantiation in a visualization. In an AVD-compliant visualization, changes in data produce *commensurately* important changes in a visualization, and vice versa: trivial changes to data should not result in significant visual changes in the visualization, whereas significant changes in the data should always be visible. I think this is a reasonable goal in line with the goal of “robustness” laid out in Allen et al. [APW*19]: a raincloud plot should allow viewers to detect meaningful differences between distributions, while avoiding highlighting trivial differences or creating visual artifacts due to sampling error or poor choices of design parameters.

In the following sections, I present defensive analyses of how raincloud components can produce these sorts of AVD errors. In some cases, it is possible to trivially produce examples of these failures—in those cases, I rely on visual examples. For others, the scope of the problem is more difficult to assess from single examples; in those cases, I employ simulation to detect the existence of algebraic errors (as in Crisan & Correll [CC21]). There are many families of distributions and a large space of designs and design parameters: I employ simulation here not to fully map this space, but to identify specific issues as they arise across common classes and scales of univariate data.

Table 1 summarizes the AVD issues discussed in

this paper. All examples and analyses in this paper are available as an interactive **Observable** notebook: <https://observablehq.com/@mcorrell/raincloud-robustness>. The plots themselves are rendered as raster images using `p5.js` to afford pixel-based measures of difference.

3.1. Hallucinators

As stated by Kindlmann & Scheidegger, per AVD [KS14]:

The Principle of Representation Invariance (or just Invariance) says that visualizations should be invariant with respect to the choice of data representation: changing the representation should not change the visualization. A visualization failing this principle has a hallucinator: a different impression was created (hallucinated, in fact) out of nothing but a different representation of the same data.

An example of a hallucinator is a multiclass scatterplot with significant overdraw: in the absence of α -blending or other reordering techniques [CEJ*18], the order that the various points are drawn can create visually distinct visualizations, despite not representing a genuine change in either the underlying data or visual representation. Given the number of design decisions involved in raincloud plots (not just what sort of components to use, but additional parameters like histogram bins or mark size or opacity), it is not surprising that hallucinators would occur in this setting as well.

A trivial example I previously discussed is that of **Interval**-based lightning designs. As shown in Figure 4, there are many potential designs that amount, visually, to a point atop a line representing some interval. The choice of *which* interval to visualize can result in lines of radically different visual length (and interpretation) despite sharing the same underlying data and representation. For instance, in my implementation of **Boxplot** and **Midgap** plots, I follow the common suggestion of Frigge et al. [FHI89] and have the whiskers denote ranges of $1.5 \cdot \text{IQR}$ from the closest quartile. However, in describing the **Moment Plot**, Potter et al. [PKRJ10] recommend using whiskers to denote the full range of the data, and so I follow their lead in my implementation. Without this admission, or an inspection of my code, a viewer could be forgiven for thinking that these designs encode different underlying data distributions.

A more complex hallucinator (and pressing, given the observed dominance of **Jittered** dot plots in rainclouds) arises from the use of random jitter in **Jittered** dot plots. This disruptive “carpet bombing” [Few17] of the position channel means that each new rendering of the plot will be visually distinct, even with no changes to the underlying data. This visual distinction makes it very difficult to identify similarities and differences between distributions. The reader is invited to try this exercise for themselves in Figure 5. Deterministic layouts of points lack this hallucinatory quality (although are vulnerable to other issues, see below).

3.2. Confusers

Again quoting Kindlmann & Scheidegger on AVD [KS14]:

The Principle of Unambiguous Data Depiction (or just

Issue	Description	Algebraic Failure	Impacted Components
Ambiguity	Statistical graphics can look visually similar but have distinct underlying meanings and methods of computation.	Hallucinator	Lightning
Randomness	Randomness in layout makes the disambiguation of similar distributions difficult.	Hallucinator	Rain
Aliasing	Multiple, radically distinct, distributions can have the similar statistical summaries.	Confuser	Lightning
Overplotting	Comparing relative density is difficult when there are too many overlapping marks.	Confuser	Rain
Oversmoothing	Kernels that are too large (or bins that are too coarse) can hide important distributional features.	Jumbler	Cloud
Discretization	Binning produces the impression that the distribution consists of discrete (rather than continuous) values, which may or may not be the case.	Jumbler	Cloud, Rain
Undersmoothing	Kernels that are too small (or bins that are too fine) can produce visual artifacts (spikes and gaps) that are not reliable distributional features.	Misleader	Cloud
Renormalization	Adjusting the scales or binning scheme of a plot is visually disruptive, but may not reflect an important change in the data.	Misleader	Cloud, Rain

Table 1: A summary of the raincloud-related AVD violations discussed in this paper.

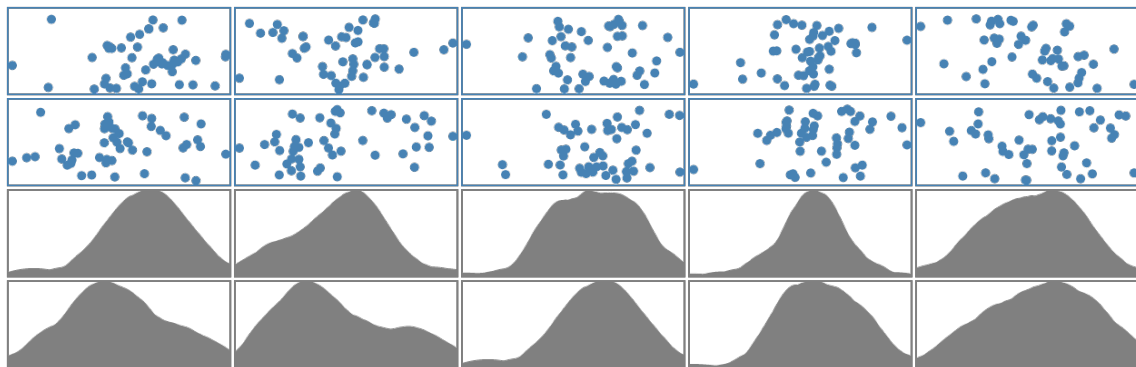


Figure 5: *Jittered* plots hallucinate visual differences between plots even with the same data. Two, and exactly two, of each “lineup” [WCHB10] of ten charts contain identical data. The other eight are different samples from the the same sampling distribution. Because jittering is randomized on a per-plot basis, it is difficult to disambiguate visual differences caused by data changes versus visual differences caused by jitter. *Density* plots reveal the identical data (the first and eighth charts in both cases) while still communicating the unimodal roughly bell-shaped nature of all ten samples.

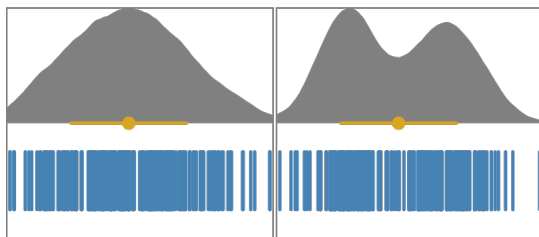


Figure 6: Both overplotting as well as choices of representation can result in confusion. Two datasets of 200 points. The data on the left are drawn from a unimodal gaussians, while the data on the right are drawn equally from two unimodal gaussians. The *Mean+Interval* does not reveal the existence of these two modes. Likewise, overplotting in the *Strip Plot* hides information about modes, as both datasets have significant visual density both near and far from their mode(s). A *Density* plot is a necessary addition to reveal the hidden shapes.

Unambiguity) says that visualizations should be unambiguous: changing the underlying data should produce

a change in the resulting visualization. Failing this principle, a visualization has confusers: changes in the data that are effectively invisible to the viewer of the visualization.

There are a number of cases where different data fail to produce visual changes in the resulting univariate visualizations. As with hallucinators above, lightning designs are perhaps the most trivial source of these issues. Choonpradub & McNeil [CM05], for instance, present sets of very distinct datasets (such as unimodal Gaussians, bimodal Gaussians, highly skewed data with a significant outlier, etc.) that all have the same five number summaries and so, by definition, all would have an identical **Boxplot**. Since lightning designs visualize summary statistics, any examples in the long tradition of datasets like Anscombe’s quarter [Ans73] that illustrate the weaknesses of these summary statistics can create a confuser.

However, I note that supplementing lightning with additional plots may not be sufficient to resolve the confuser. Figure 6 shows an example where the addition of rain is not sufficient to disambiguate data: overplotting and high data density can make distributional shape hard to recover. Likewise, the “adversarial” univariate plots in Correll et al. [CLKS19] represent attempts to intentionally

hide important distributional features by using overplotting, coarse histogram bins, and over-smoothed KDEs.

3.3. Jumlbers & Misleaders

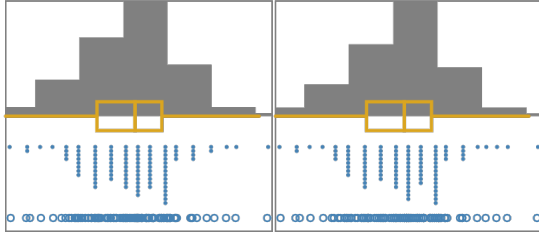


Figure 7: A *Histogram-Dot Plot-Box Plot* that results in jumbling: changes to the underlying data do not produce visual differences. On the left are 100 samples from a gaussian ($\mu = 0$, $\sigma = 20$). On the right, the same data have been converted to integers. The resulting rainclouds are almost entirely identical: the discretization of the data is not visible. The visibility of this change in the rain component, which is intended to reveal these sorts of differences in raw data, is contingent on the size of the marks, the jittering strategy, and the data density. Certain layout choices, such as Wilkinson dot plots, can even further obscure discretization, as they perform a further discretization step of their own.

A violation of the last AVD principle produces two classes of errors, again as per Kindlmann & Scheidegger [KS14]:

The Principle of Visual-Data Correspondence (or just Correspondence) says that significant changes in the data should meaningfully correspond with noticeable changes in the visual impression and vice versa. If an important change in data is not clearly manifested in the visualization, it has jumbled the data. If a clear and obvious transformation of the visualization corresponds with an unimportant change in the data, the visualization is misleading.

One concern with the Principle of Correspondence is that both “significant changes” in data and “noticeable changes” in the reading of a visualization are to some extent dependent on the viewer and their tasks. However, I note that distributional “sanity checks” [CLKS19] often rely on detecting data differences that are relatively small in scale. This means that factors such as overplotting and overaggregating can hide all number of potentially important data changes like missing data [MKC20] or changes in internal density. For instance, in Figure 7 the discretization of data is arguably a significant change, but can be difficult or impossible to detect with all three components of a raincloud, contingent on many features of both the design and the data. Visualizations in which these important data changes are difficult to detect are *jumlbers*.

The need to bin or aggregate can also create *misleaders*, where visual disruption is not indicative of any particularly relevant or important change in the data, but more an example of hitting arbitrary thresholds in the design parameters. For instance, Correll & Heer [CH17] point to the “renormalization bias” in visualizations of density information. Many charts (including all of the raincloud

or raincloud component charts in this paper) have *a priori* fixed dimensions and color scales, and so must dynamically adjust scales in order to fit all of the data. Adding a new data point (for instance, to the modal column in a Wilkinson *Dot Plot*) is a relatively minor and unimportant data change (after all, it is already the mode and remains so after the new data point is added), but can cause a visually disruptive renormalization where marks are resized and rebinned to fit in the chart bounds. A *Heatmap* presents the largest potential for a misleader: adding to the mode causes a renormalization of the entire color scale, impacting almost every pixel in the chart, whereas adding an outlier to an otherwise empty region of the distribution (arguably a more significant data change) only alters pixels in the immediate vicinity of the new data. Figure 8 shows an example of a potential misleader caused by the renormalization bias. One unique downside of raincloud plots is that these thresholds (and so subsequent visual changes) occur in ways that may not be synced across the three components of the raincloud. For instance, the number of bins in the *Histograms* shown in this paper are determined via d3’s [BOH11] standard binning function, which employs Sturges’ rule [Sco09] by default. Sturges’ rule, in turn, is sensitive only to the number of datapoints. Resizing points in a Wilkinson *Dot Plot* to fit in a given size is dependent on the maximum number of points in a particular stack of points, rather than the number of bins in total, and so dependent on local density. A visually disruptive rebinning event can therefore occur in a *Histogram* component of a raincloud plot while leaving all of the other components of the raincloud relatively untouched.

4. Discussion

The defensive analysis in this papers leads me to a set of conclusions somewhat at odds with the current use and conception of raincloud plots. I condense my findings in the subsections below, with an eye towards influencing the *design* and *use* of this relatively under-explored form, and with an eye towards locations where we (as a community) are in need of *future work* to empirically assess rainclouds and their components.

4.1. The Unsuitability of the Standard Design

The *Density-Jittered-Boxplot* raincloud, while not the only variation of the raincloud proposed in Allen et al. [APW*19], is both a common and in some sense emblematic example of the chart type, showing up everywhere from the *Xenographics* entry for rainclouds to the logo of the {rainclouds} R package. Yet, I maintain that this design has several key AVD vulnerabilities.

For one, the *Jittered* dot plot is an almost textbook example of an AVD failure: the random layout of points makes comparison of distributions difficult, and is relatively meaningless despite having large visual impact. I recommend, instead, either a more principled choice of jittering (when the number of points is small enough that individual marks are legible in such designs), or, as a reasonable compromise, a *Strip Plot*. While *Strip Plots* are prone to overplotting, I would still advocate for their use in many scenarios. For one, this overplotting is often acceptable, as overplotting is mainly problematic for hiding density information, and other components of the raincloud can present this information more directly (see 4.2). Secondly, other rain designs less prone to overplotting can still hide

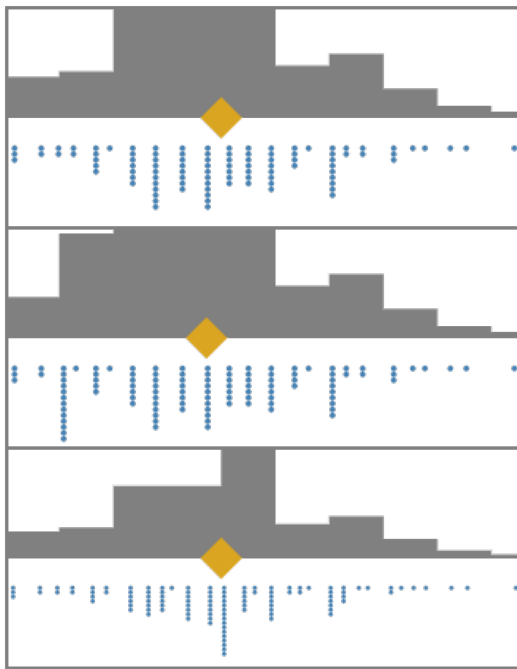


Figure 8: 100 points from a gaussian as visualized in a *Histogram-Dot Plot-Mean Marker* raincloud. Adding 10 points to the interior of the distribution ($1.5 \cdot \sigma$ from the mean) is not visually disruptive, impacting only a single bin of the *Histogram* and a single column of the *Dot Plot*, and moving the *Mean Marker* slightly to the left (middle chart). Adding 10 points to the center of the distribution, however (bottom chart), causes a “renormalization bias” [CH17]: the height of all the bins of the *Histogram* are impacted, the size of the marks in the *Dot Plot* are decreased, which in turn causes a global re-binning. This is misleading, as one would expect visual changes to be commensurate with the importance of the underlying data changes, and these data changes are roughly equivalent (if anything, adding inliers to a sparse region would seem to be more informative than adding to an existing mode).

important information about the raw values, and so choosing an alternative does not eliminate this problem. Lastly, the difficulty of estimating means or modes or relative density in overplotted *Strip Plots* might be a “beneficial difficulty” [HAS11] as viewers will not be tempted to make estimates that may be incorrect.

Another concern with the *Density-Jittered-Boxplot* raincloud is that the standard *Boxplot* has issues across multiple AVD failure types: it does not represent important distributional quantities in an unambiguous and robust way. I would suggest instead that these marks be more driven by intended tasks and analyses of the viewer, or at the very least “map” [FLCT21] to the actual statistical analyses performed. For instance, if used in a report using Bayesian analyses, a Bayesian credible *Interval* might be a better choice for *lightning*. In situations where there is no clear definition of what sort of intervals or summary statistics to expect, a simple *Marker*

for the mean might produce less ambiguity or potential for misuse or misinterpretation.

Beyond avoiding *Jittered* dot plots and *Boxplots* in rainclouds, I am less dogmatic about the choice of cloud. For discrete data, the visual metaphor of the *Histogram* might be more appropriate, as there is less temptation to assume that values are continuous than in a *Density* plot. Similarly, a *Density* plot, since the KDE can extend infinitely far in all directions, might suggest the existence of points outside of the observed (or even possible) range of values. Furthermore, there are also cases where there are existing “semantic bins” [SCB22] of meaningful partitions of the data, and so a discretization would be most appropriate or expected for the intended audience. However, these are all relatively narrow and contingent reasons to suggest the use of a *Histogram*: in general, both *Histogram* and *Density* seem to be appropriate choices of cloud.

4.2. The Importance of Mutual Defense

The existence of AVD issues in a particular raincloud component should not be taken as final proof of their non-utility. The central conceit of a *defensive* raincloud is to permit mutual support and defense between components: while one component might have issues, the raincloud as a whole may not. The lineup in Figure 5 shows how the introduction of another raincloud component (in this case, the *Density* plot) can be sufficient to resolve particular classes of AVD errors, in particular *confusers* and *jumblers* where a chart remains the same despite replacement or alteration of the underlying data. Beyond cases where data are intentionally modified to obscure [MF17], or where the scale of the data is such that drastic changes are required to produce visual changes in the estimated density or in the summary statistics, it is difficult (but not impossible!) to conceive of a data change that will not be represented in at least one of the components of a raincloud, even if that change is difficult to perceive in the final visualization (see Figure 9).

However, choices of components should be made with the idea of mutual coverage in mind. Merely having one of each component is not sufficient to guarantee a defence against AVD issues, and not all combinations of elements produce rainclouds that are equally useful (or even coherent: the reader is invited to click the “I’m feeling lucky” button in the Observable notebook connected with this paper in order to generate one of 216 possible raincloud designs possible from the components displayed in this work). For instance, *Beeswarms* produce a visual estimation of the density of a distribution; a *Violin* is therefore not an appropriate choice of cloud, as to some extent the *Beeswarm* has already “covered” the visual metaphor of a symmetric representation of estimated density, and perhaps even more thoroughly since the *Beeswarm* affords the reading of individual values. If the *Beeswarm* is legible and sufficiently representative of the density in the underlying distribution, a cloud component may not even be necessary. In that case, it would be beneficial to include a *lightning* component for indicating summary statistics that would be hard to extract from the *Beeswarm* (e.g., adversarial datasets where estimating means is error prone, as in Ondov et al. [OYK*21]).

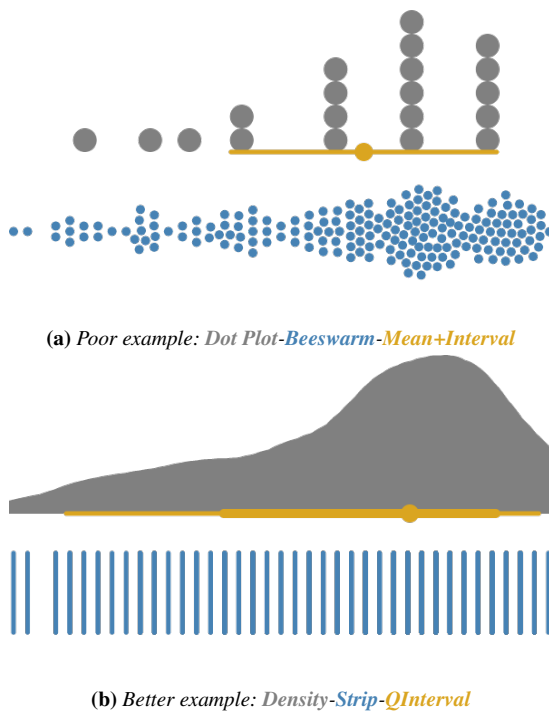


Figure 9: In a good raincloud plot, the components are mutually defensive. In these examples showing per-country life expectancy in 2012, 9a does not provide mutual defense: the *Beeswarm* provides the same information (and more!) about distributional shape as the *Quantile Dot Plot*. The individual components of 9b have deficiencies, but provides mutual defense: overplotting in the *Strip Plot* makes density hard to recover, but the *Density* plot can step in to provide it. The *Density* plot requires the user to extract central tendencies by mentally integrating the KDE, which can be difficult, but the *QInterval* explicitly encodes this information without the need for a visual estimation.

4.3. The Limitations of Rainclouds

While rainclouds do provide demonstrable benefits for visualizing distributions over their individual components, rainclouds are not panaceas. A viewer must understand the component parts, how they fit together, and be willing to accept the extra complexity (visual and otherwise) that comes with a raincloud plot. Even after these costs are incurred, there is still no guarantee that a raincloud will reveal all distributional features of potential interest (as in Figure 7). A defensive visualization is not an excuse for the designer to avoid considering the potential tasks or goals a viewer might have, nor an excuse for the viewer to avoid doing due diligence in inspecting their data.

Another limitation is not in the designs of rainclouds themselves, but in the lack of empirical work around these components, either in isolation but (especially) in how they interact. The process of moving between and among different visual representations (and visual metaphors [ZK08]) of the same data is an important but understudied assumption undergirding the use of raincloud plots. Similarly,

while there is some work in how people “sanity check” [CLKS19] and “eyeball” [BCT22] raw data, and further work in how people build up ensemble statistical pictures from raw data [SHGF16] and use that data to make inferences [KKH21], it unclear how the empirical lessons for these disparate lines of work intersect or interact. It is possible, even, that these tasks are diametrically opposed: it could be the case that a viewer looking at visualizations in order to make inferential statistical judgments does not need or want more information about the raw values in the distribution (and in fact this additional information could introduce biases if such judgments are based on inappropriate or inaccurate perceptual proxies). Likewise, a viewer interested in details about individual data values might find estimates of global density or derived summary statistics a distraction. As an example, the sample data used for Figure 1 contains two columns: not just the number of days with precipitation but also the year. A raincloud plot of the year is supremely uninteresting: there is one and only one value for each year. A viewer might wish to confirm this fact, but a raincloud in such a case is excessive when something like a binary assert or column inference, as in systems like Metareader [Jan14], would complete the job with less ink and less computation.

Another limitation and call for empirical work is with respect to the use of AVD as a tool for assessing rainclouds. While AVD can identify potential violations, a violation is not final proof of non-utility (and, conversely, the lack of violations is not proof of utility). The ultimate utility of a particular design is ultimately determined by how it is used in the real world, with real data and with real viewers. AVD can suggest potential problems or identify areas to explore in more depth, but is not a replacement for an empirical analysis of performance.

4.4. Conclusion

Rainclouds remain a promising genre of chart, able to overcome the shortcomings of existing visualizations of distributions by combining the powers and insights from multiple charts into a single view. The relative newness of rainclouds provides us an opportunity to fully explore the space of raincloud designs before the visual genre solidifies, including revisiting old designs for visualizing distributions and inventing new ones. Algebraic visualization design offers us a way of assessing this nascent design space in terms that are aligned with the mission of rainclouds: to visualize distributions without confusion, ambiguity, or waylaid by random chance. Through this algebraic lens, we find that many standard raincloud designs may provide brittle or misinterpretable summaries of distributions, and so should be designed with care, avoiding unnecessary randomness or visual designs sensitive to hyper-parameters not immediately apparent to the viewer. This notion of defensive design can extend to other design problems in visualization as well: the goal of a designer of visualization should be not just to show the data, but to make sure the data have been communicated in a safe, robust, and truthful way.

References

[ACG14] ALBERS D., CORRELL M., GLEICHER M.: Task-driven evaluation of aggregation in time series visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New

- York, NY, USA, 2014), CHI '14, Association for Computing Machinery, p. 551–560. doi:10.1145/2556288.2557200. 4
- [Ans73] ANSCOMBE F. J.: Graphs in statistical analysis. *The American Statistician* 27, 1 (1973), 17–21. doi:10.1080/00031305.1973.10478966. 1, 7
- [APW*19] ALLEN M., POGGIALI D., WHITAKER K., MARSHALL T. R., KIEVIT R. A.: Raincloud plots: a multi-platform tool for robust data visualization. *Wellcome Open Research* 4 (2019). doi:10.12688/wellcomeopenres.15191.1. 1, 2, 3, 6, 8
- [BCT22] BARTRAM L., CORRELL M., TORY M.: Untidy data: The unreasonable effectiveness of tables. *IEEE Transactions on Visualization & Computer Graphics* 28, 01 (jan 2022), 686–696. doi:10.1109/TVCG.2021.3114830. 10
- [BDL*20] BLUMENSCHNEIN M., DEBBELER L. J., LAGES N. C., RENNER B., KEIM D. A., EL-ASSADY M.: v-plots: Designing hybrid charts for the comparative analysis of data distributions. *Computer Graphics Forum* 39, 3 (2020), 565–577. doi:10.1111/cgf.14002. 1, 2
- [Ben88] BENJAMINI Y.: Opening the box of a boxplot. *The American Statistician* 42, 4 (1988), 257–262. doi:10.1080/00031305.1988.10475580. 5
- [BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D³ data-driven documents. *IEEE Transactions on Visualization & Computer Graphics* 17, 12 (2011), 2301–2309. doi:10.1109/TVCG.2011.185. 4, 8
- [CC21] CRISAN A., CORRELL M.: User ex machina: Simulation as a design probe in human-in-the-loop text analytics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2021), CHI '21, Association for Computing Machinery. doi:10.1145/3411764.3445425. 2, 6
- [CEJ*18] CHEN H., ENGLE S., JOSHI A., RAGAN E. D., YUKSEL B. F., HARRISON L.: Using animation to alleviate overdraw in multiclass scatterplot matrices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), CHI '18, Association for Computing Machinery, p. 1–12. doi:10.1145/3173574.3173991. 6
- [CF05] CUMMING G., FINCH S.: Inference by eye: confidence intervals and how to read pictures of data. *American Psychologist* 60, 2 (2005), 170. doi:10.1037/0003-066X.60.2.170. 6
- [CG14] CORRELL M., GLEICHER M.: Error bars considered harmful: Exploring alternate encodings for mean and error. *IEEE Transactions on Visualization & Computer Graphics* 20, 12 (2014), 2142–2151. doi:10.1109/TVCG.2014.2346298. 1, 3, 5
- [CH17] CORRELL M., HEER J.: Surprise! bayesian weighting for debiasing thematic maps. *IEEE Transactions on Visualization & Computer Graphics* 23, 1 (2017), 651–660. doi:10.1109/TVCG.2016.2598618. 8, 9
- [CLKS19] CORRELL M., LI M., KINDLMANN G., SCHEIDEGGER C.: Looks good to me: Visualizations as sanity checks. *IEEE Transactions on Visualization & Computer Graphics* 25, 1 (2019), 830–839. doi:10.1109/TVCG.2018.2864907. 1, 2, 4, 6, 7, 8, 10
- [CM05] CHOONPRADUB C., MCNEIL D.: Can the box plot be improved. *Songklanakarin Journal of Science and Technology* 27, 3 (2005), 649–657. 1, 2, 7
- [Dra16] DRAGICEVIC P.: Fair statistical communication in HCI. In *Modern statistical methods for HCI*. Springer, 2016, pp. 291–330. 5
- [Ekl16] EKLUND A.: Beeswarm: the bee swarm plot, an alternative to stripchart. *R package version 0.2.3* (2016). 4
- [FD81] FREEDMAN D., DIACONIS P.: On the histogram as a density estimator: L_2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 57, 4 (1981), 453–476. doi:10.1007/BF01025868. 4
- [Few08] FEW S.: Solutions to the problem of over-plotting in graphs. Visual Business Intelligence Newsletter, 2008. URL: http://mail.perceptualedge.com/articles/visual_business_intelligence/over-plotting_in_graphs.pdf. 4
- [Few17] FEW S.: The datavis jitterbug: Let's improve an old dance. Visual Business Intelligence Newsletter, 2017. http://www.perceptualedge.com/articles/visual_business_intelligence/the_datavis_jitterbug.pdf. 2, 4, 6
- [FHI89] FRIGGE M., HOAGLIN D. C., IGLEWICZ B.: Some implementations of the boxplot. *The American Statistician* 43, 1 (1989), 50–54. doi:10.1080/00031305.1989.10475612. 6
- [FLCT21] FIFE D. A., LONGO G., CORRELL M., TREMOULET P. D.: A graph for every analysis: Mapping visuals onto common analyses using flexplot. *Behavior research methods* 53, 5 (2021), 1876–1894. doi:10.3758/s13428-020-01520-2. 9
- [FWM*18] FERNANDES M., WALLS L., MUNSON S., HULLMAN J., KAY M.: Uncertainty displays using quantile dotplots or cdfs improve transit decision-making. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), CHI '18, Association for Computing Machinery, p. 1–12. doi:10.1145/3173574.3173718. 3
- [HAS11] HULLMAN J., ADAR E., SHAH P.: Benefitting infovis with visual difficulties. *IEEE Transactions on Visualization & Computer Graphics* 17, 12 (2011), 2213–2222. doi:10.1109/TVCG.2011.175. 9
- [HBO10] HEER J., BOSTOCK M., OGIEVETSKY V.: A tour through the visualization zoo. *Communications of the ACM* 53, 6 (2010), 59–67. doi:10.1145/1743546.1743567. 3
- [HKA09] HEER J., KONG N., AGRAWALA M.: Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI '09, Association for Computing Machinery, p. 1303–1312. doi:10.1145/1518701.1518897. 4
- [HN98] HINTZE J. L., NELSON R. D.: Violin plots: a box plot-density trace synergism. *The American Statistician* 52, 2 (1998), 181–184. doi:10.1080/00031305.1998.10480559. 2
- [IM87] IBREKK H., MORGAN M. G.: Graphical communication of uncertain quantities to nontechnical people. *Risk Analysis* 7, 4 (1987), 519–529. doi:10.1111/j.1539-6924.1987.tb00488.x. 3
- [Jan14] JANNAH H. M.: Metareader: A dataset meta-exploration and documentation tool, 2014. 10
- [K*08] KAMPSTRA P., ET AL.: Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of Statistical Software* 28, 1 (2008), 1–9. doi:10.18637/jss.v028.c01. 2
- [Kay21] KAY M.: ggdist: Visualizations of distributions and uncertainty. *R package version 2.4.0* (2021). 5
- [KKH21] KALE A., KAY M., HULLMAN J.: Visual reasoning strategies for effect size judgments and decisions. *IEEE Transactions on Visualization & Computer Graphics* 27, 2 (2021), 272–282. doi:10.1109/TVCG.2020.3030335. 3, 6, 10
- [KKHM16] KAY M., KOLA T., HULLMAN J. R., MUNSON S. A.: When (ish) is my bus? user-centered visualizations of uncertainty in everyday, mobile predictive systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI '16, Association for Computing Machinery, p. 5092–5103. doi:10.1145/2858036.2858558. 3
- [KS14] KINDLMANN G., SCHEIDEGGER C.: An algebraic process for visualization design. *IEEE Transactions on Visualization & Computer Graphics* 20, 12 (2014), 2181–2190. doi:10.1109/TVCG.2014.2346325. 2, 6, 8
- [MAF15] MATEJKA J., ANDERSON F., FITZMAURICE G.: Dynamic opacity optimization for scatter plots. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), pp. 2707–2710. doi:10.1145/2702123.2702585. 5
- [McN21] MCNUTT A.: What are table cartograms good for anyway? an algebraic analysis. *Computer Graphics Forum* 40, 3 (2021), 61–73. doi:https://doi.org/10.1111/cgf.14289. 2, 6

- [MF17] MATEJKA J., FITZMAURICE G.: 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* (New York, NY, USA, 2017), CHI '17, Association for Computing Machinery, p. 1290–1294. doi:10.1145/3025453.3025912. 1, 2, 9
- [MHS07] MACKINLAY J., HANRAHAN P., STOLTE C.: Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization & Computer Graphics* 13, 6 (2007), 1137–1144. doi:10.1109/TVCG.2007.70594. 5
- [MKC20] M McNUTT A., KINDLMANN G., CORRELL M.: Surfacing visualization mirages. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (2020), pp. 1–16. doi:10.1145/3313831.3376420. 2, 6, 8
- [MPOW17] MICALLEF L., PALMAS G., OULASVIRTA A., WEINKAUF T.: Towards perceptual optimization of the visual design of scatterplots. *IEEE Transactions on Visualization & Computer Graphics* 23, 6 (2017), 1588–1599. doi:10.1109/TVCG.2017.2674978. 5
- [NCE22] NEWBURGER E., CORRELL M., ELMQVIST N.: Fitting bell curves to data distributions using visualization. *IEEE Transactions on Visualization & Computer Graphics* (2022), 1–12. doi:10.1109/TVCG.2022.3210763. 5
- [OYK*21] ONDOV B. D., YANG F., KAY M., ELMQVIST N., FRANCONERI S.: Revealing perceptual proxies with adversarial examples. *IEEE Transactions on Visualization & Computer Graphics* 27, 2 (2021), 1073–1083. doi:10.1109/TVCG.2020.3030429. 5, 9
- [Phi17] PHILLIPS N. D.: Yarr! the pirate's guide to R. *APS Observer* 30 (2017). 1
- [PHKD06] POTTER K., HAGEN H., KERREN A., DANNENMANN P.: Methods for presenting statistical information: The box plot. *Visualization of Large and Unstructured Data Sets* 4 (2006), 97–106. 1, 5
- [PKRJ10] POTTER K., KNISS J., RIESENFELD R., JOHNSON C. R.: Visualizing summary statistics and uncertainty. *Computer Graphics Forum* 29, 3 (2010), 823–832. doi:10.1111/j.1467-8659.2009.01677.x. 1, 2, 5, 6
- [SB91] STOCK W. A., BEHRENS J. T.: Box, line, and midgap plots: Effects of display characteristics on the accuracy and bias of estimates of whisker length. *Journal of Educational Statistics* 16, 1 (1991), 1–20. doi:10.3102/10769986016001001. 5
- [SCB22] SETLUR V., CORRELL M., BATTERSBY S.: Oscar: A semantic-based data binning approach. In *2022 IEEE Visualization and Visual Analytics (VIS)* (2022), pp. 100–104. doi:10.1109/VIS54862.2022.00029. 9
- [Sco09] SCOTT D. W.: Sturges' rule. *Wiley Interdisciplinary Reviews: Computational Statistics* 1, 3 (2009), 303–306. doi:10.1002/wics.35. 4, 8
- [She04] SHEATHER S. J.: Density estimation. *Statistical Science* (2004), 588–597. doi:10.1214/088342304000000297. 4
- [SHGF16] SZAFIR D. A., HAROZ S., GLEICHER M., FRANCONERI S.: Four types of ensemble coding in data visualizations. *Journal of Vision* 16, 5 (03 2016), 11–11. doi:10.1167/16.5.11. 2, 4, 10
- [SMWH17] SATYANARAYAN A., MORITZ D., WONGSUPHASAWAT K., HEER J.: Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization & Computer Graphics* 23, 1 (2017), 341–350. doi:10.1109/TVCG.2016.2599030. 5
- [SMY*05] SAITO T., MIYAMURA H., YAMAMOTO M., SAITO H., HOSHIYA Y., KASEDA T.: Two-tone pseudo coloring: compact visualization for one-dimensional data. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* (2005), pp. 173–180. doi:10.1109/INFVIS.2005.1532144. 4
- [Sto06] STOKES T.: Charmed, I'm sure. *Nature* 444, 7117 (2006), 264–265. 1
- [WCHB10] WICKHAM H., COOK D., HOFMANN H., BUJA A.: Graphical inference for infovis. *IEEE Transactions on Visualization & Computer Graphics* 16, 6 (2010), 973–979. doi:10.1109/TVCG.2010.161. 7
- [Wil99] WILKINSON L.: Dot plots. *The American Statistician* 53, 3 (1999), 276–281. doi:10.1080/00031305.1999.10474474. 2, 3, 4
- [WQM*17] WONGSUPHASAWAT K., QU Z., MORITZ D., CHANG R., OUK F., ANAND A., MACKINLAY J., HOWE B., HEER J.: Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), pp. 2648–2659. doi:10.1145/3025453.3025768. 5
- [YHF19] YUAN L., HAROZ S., FRANCONERI S.: Perceptual proxies for extracting averages in data visualizations. *Psychonomic Bulletin & Review* 26, 2 (2019), 669–676. doi:10.3758/s13423-018-1525-7. 2, 5
- [ZK08] ZIEMKIEWICZ C., KOSARA R.: The shaping of information by visual metaphors. *IEEE Transactions on Visualization & Computer Graphics* 14, 6 (2008), 1269–1276. doi:10.1109/TVCG.2008.171. 10
- [ZZK18] 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), pp. 1–12. doi:10.1145/3173574.3174053. 6