Nonlinear Dot Plots

Nils Rodrigues and Daniel Weiskopf, *Member, IEEE Computer Society*

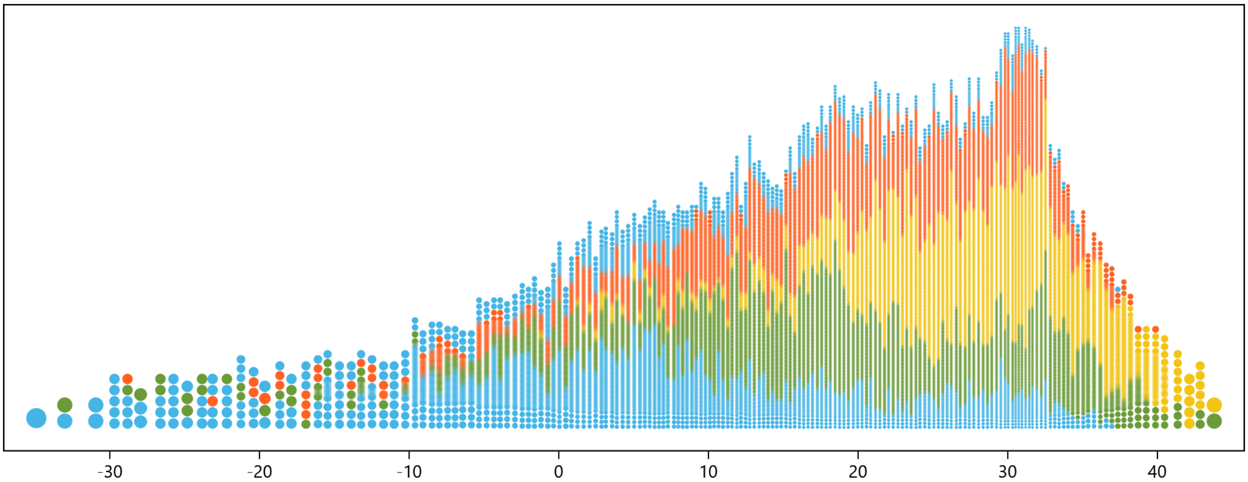


Fig. 1. A root dot plot containing more than 9,000 data points. The data set shows the monthly average of daily maximum temperatures, measured by weather stations from around the world. Individual dots are colored according to the month they represent: J F M A M J J A S O N D

Abstract—Conventional dot plots use a constant dot size and are typically applied to show the frequency distribution of small data sets. Unfortunately, they are not designed for a high dynamic range of frequencies. We address this problem by introducing nonlinear dot plots. Adopting the idea of nonlinear scaling from logarithmic bar charts, our plots allow for dots of varying size so that columns with a large number of samples are reduced in height. For the construction of these diagrams, we introduce an efficient two-way sweep algorithm that leads to a dense and symmetrical layout. We compensate aliasing artifacts at high dot densities by a specifically designed low-pass filtering method. Examples of nonlinear dot plots are compared to conventional dot plots as well as linear and logarithmic histograms. Finally, we include feedback from an expert review.

Index Terms—Nonlinear dot plot, statistical graphics, sweep algorithm, layout

1 Introduction

Dot plots [36] provide a good representation of the frequency of data items, serving the same goal as bar charts that show histograms. They stack data points in the form of vertical columns of constant-size dots along a horizontal axis. Like regular histograms, they use the horizontal axis to show data values and the vertical axis for the frequency. However, there are important differences between dot plots and histograms. First, dot plots use a separate dot for each data sample; therefore, it is very easy and intuitive to actually count data items. Second, dot plots do not use fixed uniform binning along the data value axis but rather put dots at the exact horizontal position that corresponds to the data value. Therefore, dot plots tend to be a more intuitively understandable, clearly readable, and accurate representation of the distribution of data values than histograms.

Unfortunately, dot plots do not scale well to very large data sets, especially with a high dynamic range of frequencies. In particular, outliers become very difficult to detect, as they will be rendered as minute dots in seemingly empty spaces. Small gaps in areas with a high dot density are smoothed out because of the constant dot size, and therefore become unperceivable.

We address this problem of high dynamic frequency range by extending conventional dot plots to *nonlinear dot plots*. The basic idea is to combine the concept of the dot plot with the nonlinear scaling as known from logarithmic histograms (i.e., with a logarithmic scaling of bar heights). To this end, we have to loosen the constraint of a constant dot size. Now, the dot radius is scaled according to the height of the dot column. Unfortunately, this scaling has implications that make its realization more difficult in comparison to bar charts: Since dot plots do not use uniform binning, the dot radius affects the positioning of the dots; and scaling the radius does not only change the column height but also its width. Therefore, the overall layout of the dot plot needs to be dynamically adapted. Figure 1 shows an example of a nonlinear dot plot.

Our main contribution is the model of data-adaptive scaling of dots in combination with a new layout algorithm that supports this nonlinear scaling. It is a two-way sweep algorithm that traverses the data value range (horizontal axis of the diagram) in both directions. In this way, we can guarantee a dense and symmetrical layout with a computational complexity of $O(n)$ for n data items. A sample implementation of our algorithm is publicly available on GitHub[1]. The second contribution is an improved rendering method for dot plots that allows us to remove aliasing artifacts at high dot densities by low-pass filtering along dot columns.

Nonlinear dot plots include conventional, linear dot plots as a special case (of constant dot size). Therefore, even linear dot plots benefit from our improved two-way sweep algorithm. However, the main

• *The authors are with VISUS, University of Stuttgart, Germany. E-mail: nils.rodrigues@visus.uni-stuttgart.de, weiskopf@visus.uni-stuttgart.de.*

Manuscript received 31 Mar. 2017; accepted 1 Aug. 2017.
Date of publication 28 Aug. 2017; date of current version 1 Oct. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2017.2744018

[1] https://github.com/NilsRodrigues/nonlineardotplots

benefit is the flexibility in adjusting the dot size, facilitating the accurate visualization of distributions with a wide range of frequencies. For example, they can resemble semi-log plots of histograms in which the x-axis is unaltered, but the y-axis is scaled logarithmically. We also show that (nonlinear) dot plots can be easily combined with per-dot color-coding of additional data attributes, and integrated with other variants of statistical plots, such as box plots. We compare examples of nonlinear dot plots to linear dot plots as well as linear and logarithmic histograms to show where they can be best used. This evaluation is complemented by feedback from an expert review.

2 Related Work

Dot plots have a long tradition as a type of diagram used for statistical graphics. Jevons used them as early as 130 years ago [18].

Cleveland describes a modern use of the term "dot plot" [6]. He proposes bar charts with less visual clutter, by replacing the bars with only single dots at their end, directly depicting the value. Cleveland also presents multiway dot plots [7], in which he uses round markers in X-Y-plots. These types of visualization are often used in statistical contexts and have proven to be well suited for small data sets [2, 24]. They typically use a nominal scale for one dimension and place multiple data series in the same plot. While Cleveland uses dots in his visualizations, they are still based on, and very related to, other types of plots. Another diagram type that uses dots are histodot plots [36], which apply uniform binning to place the dots, but are fundamentally histograms with different rendering; this approach is used, for instance, in Sasieni and Royston's plots for statistical analysis [25].

While all of the above techniques use dots in the context of frequency visualization, we do not consider them as "dot plots" in the strict sense. In fact, we follow the specification by Wilkinson [36], who describes and analyzes dot plots in great detail. He also shows the differences between actual dot plots and histodot plots. He describes a number of rules that apply to dot plots: (1) Every data point has a corresponding dot; (2) all dots are of the same size; (3) dots are not merged, blurred, stretched, or squashed. One implication is that the dot size is indirectly determined by the size of the overall diagram, its aspect ratio, and the number of dots. Even with the above constraints, there is a variety of conceivable layout algorithms. A recent approach by Dang and Wilkinson [10] enhances dot plots by producing symmetrical visualizations. The layout algorithm first draws the highest columns and then centers them on the included data points. This creates plot areas that are symmetric if the underlying samples also exhibit symmetry. Therefore, the layout better characterizes the source data.

Dot plots are useful for relatively small or low dynamic range data sets. They serve for finding clusters and gaps as they show the frequency or density of values directly. At the same time, they also show outliers at the exact location corresponding to their value. The simplicity of the plots themselves and the countability of individual dots makes this visualization intuitive to understand [3]. According to Padmi and Russasmita [22], even children with no prior experience can quickly learn and interpret dot plots, gaining insight into the characteristics of displayed population data. The layout algorithm is simple enough for students to draw the plot by hand (i.e., without computer hardware) and use their results for ad-hoc analyses. There are other indications why dots can be advantageous over other visual representations: For example, Dang and Wilkinson [10] discuss advantages in the context of stacking elements; Tory et al. [29] showed with a user study that dot displays are more effective than landscape visualizations for remembering spatialization.

With this paper, we want to build on the existing work on dot plots and exploit their effectiveness in visualizing data distributions. To our knowledge, there is no dot plot variant that would allow for varying dot size within one diagram. Therefore, nonlinear scaling for high dynamic ranges is not possible yet. This is the main problem that we want to address with this paper.

The general problem of defining adequate dot sizes is related to the problem of finding a suitable bin width for histograms or the bandwidth in kernel density estimation (KDE). Scott [26] introduces a method for selecting an optimal bin width based on the number of samples and the distribution's standard deviation. Wand [34] extends Scott's approach to fit more types of data. Interpreting a histogram not only as a visualization but as the frequency distribution of underlying samples, allows using computational methods for various adaptive image equalizations [23, 37, 38]. All of the above techniques find a single, yet data-dependent width. There are other approaches that allow for adaptation within a data set. For example, adaptive histograms (as in the "ahist" function of the Ckmeans.1d.dp [35] package for the R environment) aim at the visualization of data that contains multiple clusters, with low dynamic range in each cluster; they create a single visualization by combining histograms with fixed width per sample cluster. As another example, Copin et al. [9] use adaptive spatial binning with a quadtree for their 2D renderings. However, none of these approaches target the problem of dot plots. In particular, most of the existing binning approaches lead to an (adaptive) partitioning of the space of data values, whereas nonlinear dot plots differ in the sense that they leave potentially empty space between columns of dots.

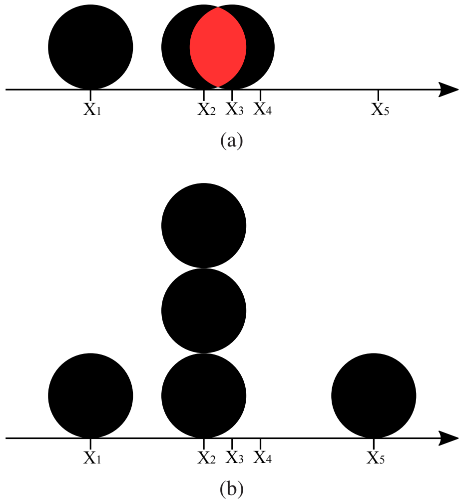


Fig. 2. Illustration of Wilkinson's sweep layout algorithm [36]. Adding a dot for x_3 would create an overlap (marked red) with the previously placed dot at x_2 (a), and similarly for x_4. These dots are therefore stacked (b). x_1 and x_5 can be plotted without any issues.

With nonlinear dot plots, the dot size decreases when multiple data points are close to each other. This can be taken to the extreme: keep the column height constant but only adapt the dot size. If the data density is high enough, the result resembles a mixture of jittered strip charts and stripe charts [5]. They are, however, interrupted by columns containing only one or two dots that are rendered relatively large and cover up areas that would otherwise be empty spaces.

The general problem of dealing with large or high dynamic range data has also been an issue for visualizations other than dot plots [21]. For instance, histograms and bar charts in general, can be adapted to use logarithmic scaling. This makes the graphics harder to interpret [1, 4, 8, 16, 30], but allows displaying the data without significant changes to rendering algorithms. Nonlinear dot plots can also be seen as using a different but linear height scale per column, thus composing a multi-scale visualization as described by Isenberg et al. [17].

3 Background of Linear Dot Plots

We use a simplified version of Wilkinson's original algorithm for linear dot plots [36] as a starting point for our extension to nonlinear dot plots. For a self-contained description, we briefly summarize the original algorithm without smoothing and lateral offsets in this section.

As illustrated in Fig. 2, Algorithm 1 performs an upward sweep through the data to place the dots from left to right. It assumes that the data samples are given as data points with corresponding data value x_i, with index $i = 1, 2, \ldots, n$ for n samples in the entire data set. It depends on the distance d, which is the (constant) diameter of the dots for the creation of columns. The key point is that the algorithm keeps

1. Sort input data values ascending
2. Take lowest value x_i that has not been plotted
3. Start with $c = 1$
   Increment c while $|x_i - x_{i+c}| \leq d$
4. Place $|c|$ dots above x_i in layout
5. Mark values x_i to x_{i+c} as plotted
6. Repeat until there are no data values left

Algorithm 1. Wilkinson's original sweep algorithm

1. Sort input data values ascending
2. Take **lowest** / **highest** value x_i that has not been plotted
3. Start with $c = \pm 1$
   Increment / **Decrement** c while $|x_i - x_{i+c}| \leq d(c)$
4. Place $|c|$ dots above x_i in layout
5. Mark values x_i to x_{i+c} as plotted
6. Repeat until there are no data values left
7. Do one **upward** and one **downward** pass
8. Use average of **upward** & **downward** pass

Algorithm 2. Adaptation for symmetric, nonlinear dot plots.

incrementing the number of dots, c, for the current column as long as further dots still overlap with the column (step 3 of Algorithm 1).

Dot plots link the number of data points—which has to equal the number of dots—and the size and aspect ratio of the diagram with the dot size. Wilkinson recommends a default diameter of

$$d = \sqrt[2]{n}\,, \tag{1}$$

where n is the number of data points. He also recommends an aspect ratio of 5:1 for the entire diagram. If the dot columns become too high and overplotting occurs, the dot size needs to be reduced in order to preserve the aspect ratio.

4 Visualization Technique

To accommodate large, high dynamic range data sets and preserve the advantages of dot plots, the diameter has to be varied in the same plot: the higher the dot column, the smaller the rendered symbols. This leads to a nonlinear modification in column height and width, which is not very intuitive to interpret. In a histogram with nonlinear scaling, we would only have to measure the height and know the transformation function in order to calculate the represented value. With nonlinear dot plots, we also have to factor in varying column widths, making the computation of displayed value densities more difficult (see Section 4.4). However, the total size of the plot can be reduced, while retaining large dots for outliers. The decrease in height also provides enough flexibility to bring the output closer to the optimal aspect ratio.

We first describe an extended two-way sweep algorithm that allows us to work with varying dot sizes (Section 4.1). Then, we discuss the merging of intermediate results of the two-way sweep (Section 4.2), models for the data-driven adaptation of the dot diameter (Section 4.3), resulting envelopes (Section 4.4), and aliasing from rendering dots (Section 4.5). The section closes with extensions and variants of the visualization, including color-coding and overlays with other diagram components (Section 4.6).

4.1 Two-Way Sweep Algorithm

Our new two-way sweep algorithm is summarized in Algorithm 2. We place it side-by-side with Wilkinson's original algorithm (without additional smoothing) to clarify the modifications and extensions.

To adapt the dot size we have to replace the constant diameter from Wilkinson's original algorithm (step 3) by a data-dependent variant:

$$d(c) = d_{\text{single}} \cdot f(|c|)\,, \tag{2}$$

where $|c|$ is the current number of data points in a column and d_{single} is a constant start diameter that also facilitates an overall scaling of all dots. The function $f(c)$ should be weakly monotonically decreasing, i.e., dots should become smaller for columns with more data points. Also, we should have $f(1) = 1$, so that we obtain the diameter d_{single} if just a single dot is placed. Section 4.3 discusses concrete examples of $f(c)$.

As indicated in Algorithm 2, we can use the data-dependent dot diameter (Eq. (2)) within Wilkinson's original single-sweep algorithm. However, while this single-sweep approach works very well for a constant dot size, it exhibits some issues when the data value density decreases along the sweep direction. The underlying reason is that shrinking dot size (during the sweep) should have an impact on which dots should be included in, or excluded from, a column.

To illustrate this problem let us assume that the data consists of a dense, sorted group of values X. The first and last value are within the initial dot diameter, which would lead to rendering a single column for linear plots: $x_n - x_1 \lessapprox d$. During the sweep from the smallest to the largest value x_i, the first column becomes narrower: $d(c_1) \ll d$. Only a few values from X remain and create their own column, with only little less width than the initial diameter: $d(c_2) \lessapprox d$. Since all values have less than d distance, there will be overlap between the two columns' dots.

The inverted case, however, presents no problems: When the density increases along the sweep direction, the columns become narrower. Therefore, a second column will not be as wide as the first one and there will not be any overlap between them.

This observation leads to the solution of the problem: we use two sweeps—one in each direction—and combine their results by averaging. Algorithm 2 indicates the two sweeps as "upward pass" and "downward pass". For the "downward pass", the index variable c has to be negative. However, $|c|$ denotes the current number of dots in the column, regardless of the sweep direction. Section 4.2 describes in more detail how the two sweeps are combined.

The original sweep algorithm only needs one pass over the source data. Since none of our alterations are of higher complexity than $O(n)$, our proposed algorithm still has the same linear runtime complexity (with a factor of roughly 2 for the number of computations).

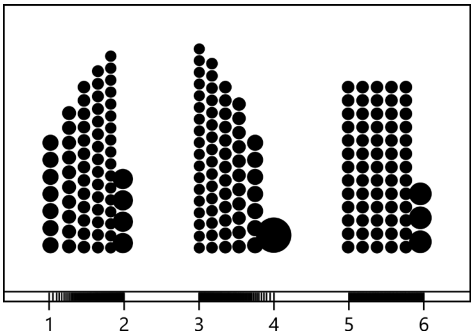
4.2 Combining Sweeps

Dang and Wilkinson's greedy version of a dot plot algorithm [10] was designed to achieve symmetrical diagrams. While it is preferable to create symmetrical visualizations when using symmetrical data, the algorithm either introduces severe overlaps or gaps between dot stacks (depending on the chosen factor for h in step 1). As already indicated in Section 4.1, a single sweep direction creates overlap problems for nonlinear dot plots, too. In addition, it also causes asymmetry under certain conditions.

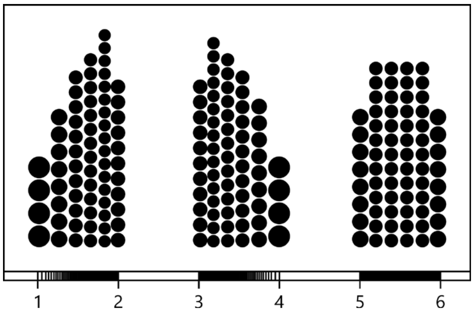
To address these problems, we decrease the sweep direction's influence on the final layout by combining both directions, as described in Algorithm 2. The open question is: how are the results of two sweeps adequately combined?

As we will explain later on, the rules we imposed on the dot diameters (Section 4.1) guarantee that both sweeps return the same number of stacks. We can therefore also define a one-to-one mapping of columns from one layout pass to the other. Columns resulting from a single sweep are defined by three values: a position in the data dimension as well as the number and diameter of the dots. For the merging of stacks from opposite sweep directions, their positions can be readily averaged by using their arithmetic mean. However, it is not straightforward to merge the actual dots if the two columns contain a different number of them: the arithmetic mean of the number of included dots might contain "half" points, but we can only draw entire dots.

We choose to only draw entire dots (by rounding off the number of dots in the current column) and carry the remainder to the following



(a) Single left-to-right pass



(b) Two opposed passes combined

Fig. 3. Two dot plots of the same data with varying value density: rising from 1–2, falling from 3–4, and constant between 5–6. Vertical lines on the bottom axis show the individual data value positions. Merging two passes results in a plot with less overlap and more symmetry (b).

column. In total, this will result in the same number of dots as from a single pass and the remainder is only distributed within a neighborhood of columns with less than d_{single} distance. The reason for this is that data values that are further apart than d_{single}, form clusters that do not interfere with each other in the sweep algorithms. Therefore, dots from different clusters can never appear in a column together.

In summary, the arithmetic mean can be applied to the column's positions directly and their height (number of dots) with the slight modification above. However, the arithmetic mean is not suitable for the diameter of rendered dots. This is an implication of the nonlinear dependency between c and $d(c)$ in Eq. (2). Instead, we calculate the correct diameter from the actual number of dots in each column by applying Eq. (2) to c after the number of dots is averaged.

There remains one prerequisite for the two-pass method to work: both passes must return the same number of columns. This is important because the first column of the upward pass has to be averaged with the last column of the downward pass. If they do not match, the data values of different neighborhoods will be merged. Let us analyze the different scenarios encountered during the layout process, to check whether the above prerequisite can be met. Our first observation is that data values that are further apart than the start diameter d_{single} divide the data samples into clusters that can be treated separately. Therefore, we can restrict the discussion to a single cluster of data samples.

Clusters that produce a single column in one direction will also produce a single column in the opposite direction, i.e., this is a trivial case. Let us assume the upward pass A resulted in two columns and look at the following two scenarios.

Scenario 1: If the downward pass B returned a single column, it would mean that the total number of values in the cluster was so small that $d(c)$ was still larger than the distance between the first and last data value. This would, in turn, also mean that pass A would have returned a single column containing the complete cluster, which is in direct contradiction to our premise.

Scenario 2: Can pass B return three columns? If it started at the same data point, where pass A finished, then its first column would cover at least the same amount of data values as the last column of A. That would leave the same data that pass A used for its first column, which in turn would also only result in a single column of pass B. Therefore, three columns could only be generated if pass B encountered additional data values outside the cluster. However, we only look at the data inside the cluster because different clusters are so far apart that they do not interfere with each other.

According to this logic (similarly to mathematical induction), the two opposed passes cannot result in a different dot stack count, which makes our averaging method applicable to arbitrary source data.

As shown in Fig. 3b, the two-way layout leads to columns that are better centered on contained data and have less severe overlaps than the single-pass algorithm (see Fig. 3a). This makes our algorithm well suited not only for nonlinear dot plots, but it also improves on the layout of traditional, linear plots. Figure 3b also shows that a constant sample density does not necessarily lead to a constant dot stack height, which makes the visualization inconsistent with the underlying data. This is due to rendering individual dots, which leads to quantized column heights. In order to get a constant height among all columns of the third cluster, we have to fit the initial dot diameter and the nonlinear transformation to the specific samples. While these adapted parameters will fit the targeted cluster, they might be wrong for the clusters at 1 and 3. Therefore, it is not trivial to find a globally optimal solution.

It is not even clear how to completely define optimality of a solution. We might define on optimal solution as the one in which the distance of rendered dots from their input values (layout error) is minimal. If we then selected a sufficiently small dot diameter, we would render each dot at the exact data value position and have an optimal solution according to that metric. However, the resulting plot would not be readable because the dots would be too small to perceive. Therefore, an objective function for an optimal solution would have to take into account the layout error but also the dot sizes, display resolution, viewing distance, and aspect ratio. We leave the definition of such an objective function as an open question for future research.

4.3 Dot Diameter

The above two-way sweep algorithm makes heavy use of the dynamic adaption of dot size according to Eq. (2). Let us now have a closer look at useful choices for the adaptation model, as formalized by $f(c)$. For this, we assume that we have a rather dense packing of dots so that the dot plot resembles the corresponding histogram.

Although $f(c)$ controls the nonlinearity of the dot plot, it is not identical to the nonlinear mapping known from histograms or other function plots. It is important to note that $f(c)$ does not map original height for a plot to the nonlinear modification. For example, we cannot just use $f(c) = \log(c)$ to obtain the analog of a log-scale dot plot. In fact, the height of a column with c dots is:

$$h(c) = c \cdot d(c) = d_{\text{single}} \cdot c \cdot f(c) \, . \qquad (3)$$

As noted earlier, we require that $f(1) = 1$ and that $f(c)$ decreases with increasing c. In addition, we want to guarantee weak monotonicity of the plot: a column with more data points should never be smaller than a column with fewer points, i.e.,

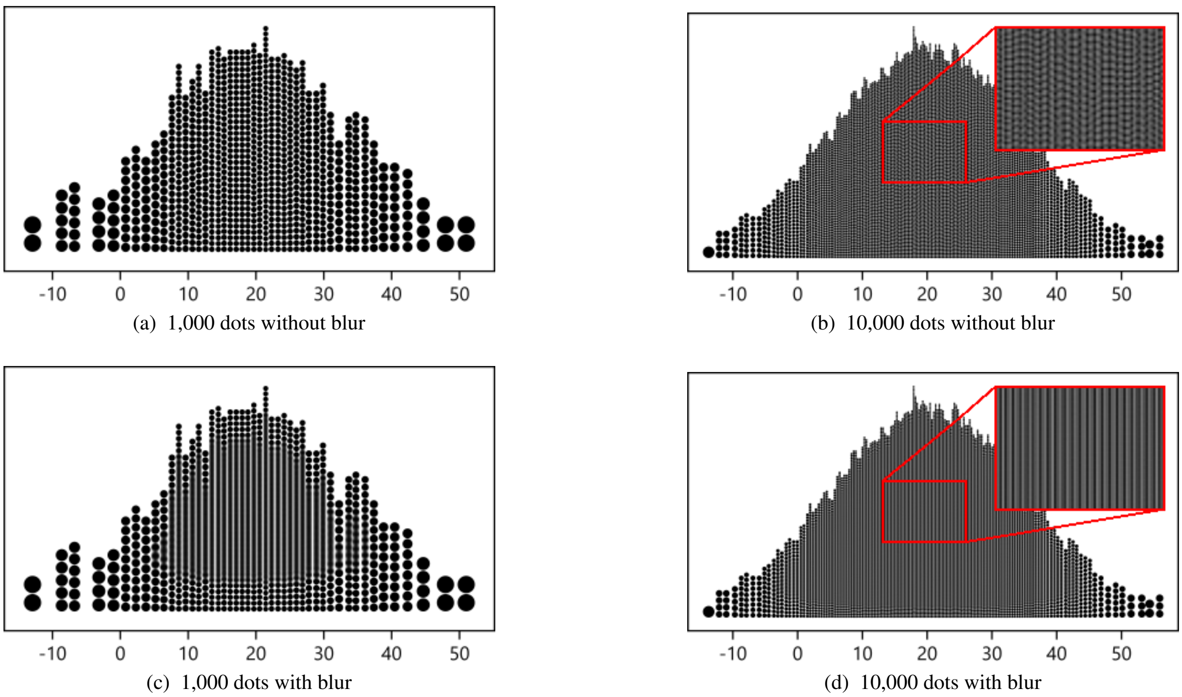
$$h(c_1) \leq h(c_2) \quad \text{if } c_1 < c_2 \, . \qquad (4)$$

The extreme case of constant height is obtained for:

$$f(c) = \frac{1}{c} \, , \qquad (5)$$

i.e., this choice leads to $h(c_1) = h(c_2) = d_{\text{single}}$. As shown later in Fig. 5b, there are some applications for this extreme model. The corresponding visualization resembles a combination of jittered strip charts and stripe charts [5] (see Fig. 5b). Typical nonlinear mappings target strong monotonicity. To this end, the dot size needs to shrink less quickly than in Eq. (5). A corresponding mapping is achieved by:

$$f(c) = \frac{1}{c^s} \, , \qquad (6)$$



(a) 1,000 dots without blur
(b) 10,000 dots without blur
(c) 1,000 dots with blur
(d) 10,000 dots with blur

Fig. 4. Low-resolution plots of normal distributions. Column-oriented anti-aliasing dampens moiré patterns in visualizations (c) and (d). When used with large dots, this blurring introduces unwanted optical effects (c).

with an additional parameter s that controls the shrink rate. The useful range for the shrink rate s is between 0 and 1. Selecting a shrink rate of $s = 0$ will create traditional linear dot plots; $s = 1$ yields the earlier model of Eq. (5). For in-between shrink rates, the column height is proportional to c^{1-s}, with the fractional exponent $1 - s$ corresponding to a root. For later example images, we selected a default value of 0.4, which leads to a column height of $c^{0.6}$. This choice led to good results for the examples of this paper, but could certainly be replaced for other examples. We added a table with varying d_{single} and s as supplemental material to illustrate the effect of these parameters on the final layout.

The root mapping in Eq. (6) might be good for many data sets, but it will not fit most processes in nature that exhibit exponential growth, where the growth rate is proportional to the current function value. Examples of such can be found in nuclear decay, the Weber-Fechner law [11], and population growth. The existence of exponential processes is also reflected in the way we represent floating point numbers: mantissa and exponent. To get a log-scale mapping, we naively try to approximate

$$h(c) = \log(c)\ . \tag{7}$$

Based on Eq. (3), we would calculate

$$f(c) = \frac{\log_b(c)}{d_{\text{single}} \cdot c} \tag{8}$$

for any desired base b, but this would violate the requirements for single dot columns, as $\log(1) = 0$. To get $f(1) = 1$, we could change the numerator in Eq. (8) to

$$\log_b(c) + d_{\text{single}}\ .$$

This, in turn, would lead to other problems. For instance, with $b = 2$ and $d_{\text{single}} = 0.5$, we would get larger dots in columns of two than in single value stacks ($f(2) = 1.5$). Instead, for logarithmic dot plots we propose

$$\log_b(c + b - 1) \cdot d_{\text{single}}\ , \tag{9}$$

as it fits all requirements when used with $b \geq \frac{\sqrt{5}+1}{2}$ (the golden ratio). The supplemental material includes a table with varying d_{single} and b to illustrate the effect of these parameters on the layout.

The final issue is the choice of d_{single}. This parameter should be chosen according to the number of data samples, their distribution, the size of the plot, and—very importantly—the targeted aspect ratio. We iteratively optimize for d_{single}: a plot is computed with the current value of d_{single}; this value is increased if the current aspect ratio is wider than the target ratio and vice versa, until the target is reached.

4.4 Envelope

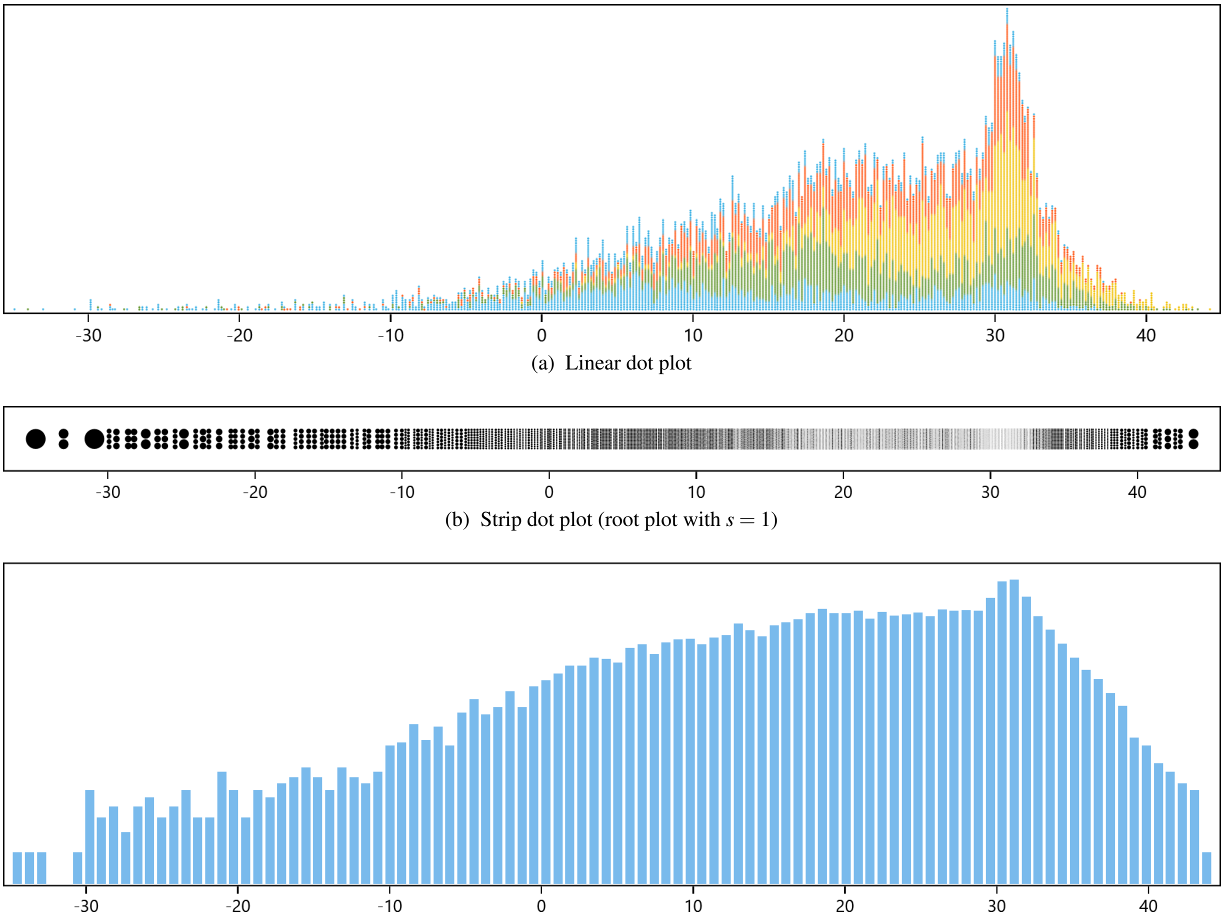
The above discussion holds for the mapping of height, i.e., a single dimension. Now, the dot plots intrinsically link column height and width because both dimensions are determined by the dot diameter. If the diameter is scaled by a factor a, the actually covered area (i.e., the area of the column) is scaled by a factor a^2. Therefore, a square root computation needs to be included in all mappings if the adjustment is meant for areas, not just height.

Based on these observations, one can consider the limit case of a very large number of dots and the envelope of the nonlinear dot plot. For the case of root dot plots according to Eq. (6), the height of the envelope scales with $\alpha^{(1-s)/(1+s)}$ if the number of dots is multiplied by α. Thus, the exponent s from the diameter scaling matches an exponent of $(1-s)/(1+s)$ in the corresponding nonlinear histogram.

4.5 Anti-Aliasing

In general, dot plots can come with high demands regarding render quality because they consist of clearly defined dots with sharp boundaries. Such boundaries in combination with some rather regular placement of dots can lead to aliasing and moiré artifacts [13,27]; see Fig. 4. The main cause of moiré effects are small differences in dot sizes. These accumulate along the height of neighboring columns and create virtual tilted lines.

Typical anti-aliasing approaches from computer graphics work with supersampling on the image plane, followed by low-pass filtering and downsampling. Our solution adopts the same strategy but exploits the special characteristics of dot plots. Low-pass filtering blurs the image, i.e., the individual dots would eventually disappear and only a solid colored area would appear. Other than generic anti-aliasing techniques, we limit blurring to the vertical direction because individual columns



(a) Linear dot plot

(b) Strip dot plot (root plot with $s = 1$)

(c) Logarithmic histogram

Fig. 5. Visualizations of the multi-annual mean of maximum daily air temperatures of each month (in degrees Celsius). Image (a) is color-coded according to the month J F M A M J J A S O N D. We use black dots in plot (b) to maximize the perceivable range of brightness.

should still be distinguishable. Furthermore, a few dots at the top and bottom are left unchanged, as they play a key role in estimating the size of individual dots and in comparing column heights.

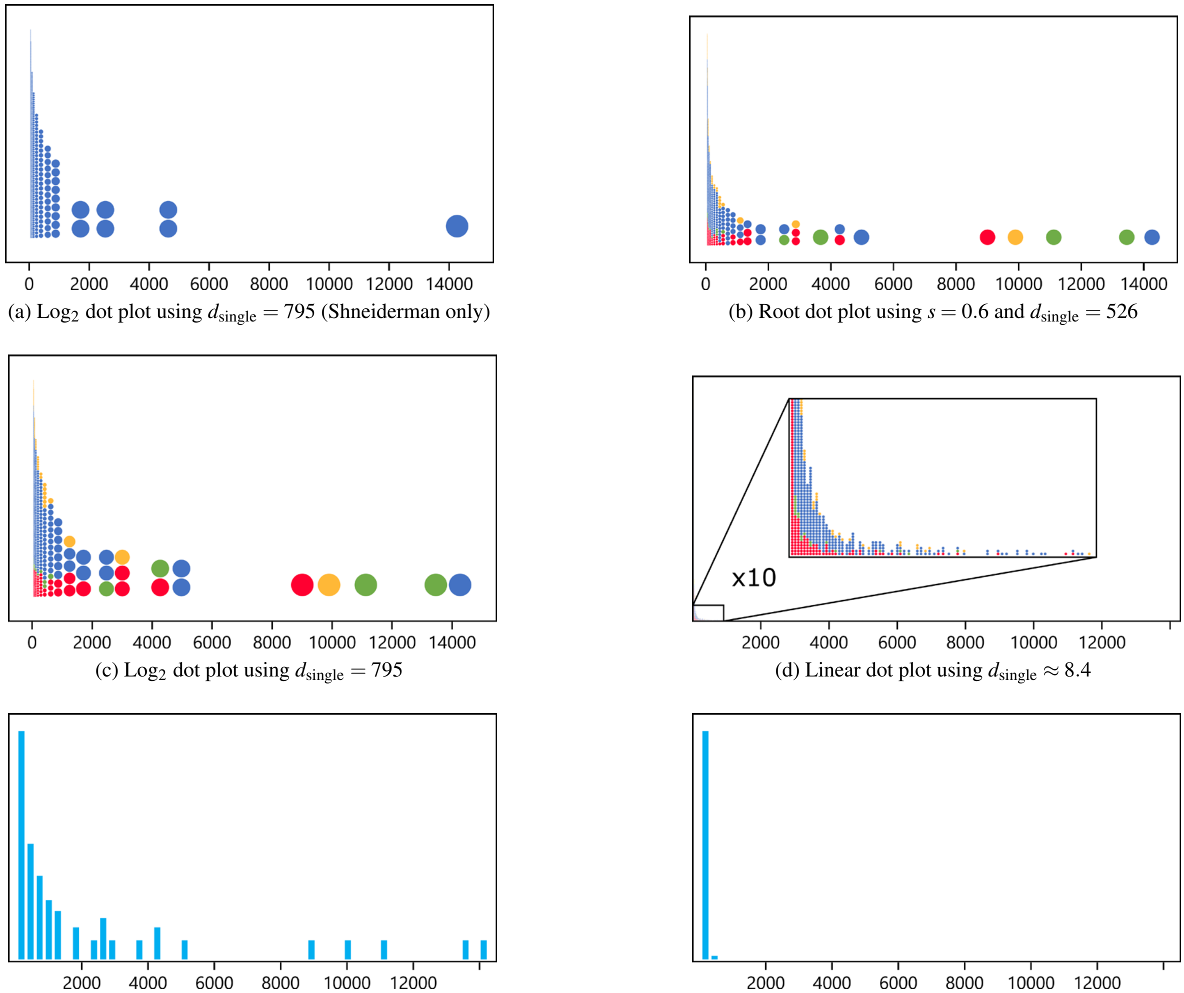
We also decided on not blurring those columns that are not surrounded by others because they do not add to the moiré effect. Finally, we only start blurring after the dot count inside a column exceeds a certain lower threshold (in our examples: 12) because otherwise, the rendering does not create an area that is big enough for the effect to be perceivable. Figures 4c and 4d show examples of anti-aliasing. Dots in a blurred line are not countable anymore. However, as the height of columns increases and the dot diameter decreases, it becomes more and more difficult to make out individual dots for counting, anyway. We use our anti-aliasing method with moderation in order to balance advantages and side effects. The vertical lines in Fig. 4c are an example of too aggressive blurring for the low dot density that trades the moiré pattern for even worse optical effects, especially when rotating the image. Therefore, we recommend anti-aliasing only for plots with very small dots as in Fig. 4d.

4.6 Variants, Extensions, and Hybrid Visualizations

Just like conventional dot plots, our nonlinear generalization can be widely applied to depict any kind of data distribution. Similarly, it can be combined with other visual mappings to include further information or emphasize certain aspects of the data.

One example is additional color mapping. In general, color plays an important role in visualization because it can show additional data attributes on top of the positional variables of the diagram. We argue that color mapping is especially useful in the context of dot plots because each single data sample generates exactly one dot, i.e., we can have a direct mapping between data sample and color. To make use of perceptual grouping by color (hue), similar colors should be spatially grouped in the dot plot. We cannot change the layout between the columns in the dot plot because they are driven by the distribution of data values. However, we may modify the order of dots within a column. Therefore, the dots in each column should be ordered vertically according to the additional data attribute mapped to color. A typical example is a chronological data distribution, i.e., a data set with samples that do not only carry some data value but also a timestamp. Such time-series distributions are best ordered chronologically in each dot column. Another possible application of color is comparative visualization of several data sets integrated in one dot plot: the color indicates the data sources. The figures in Section 5 use such colored dot plots.

We mostly render our dot plots with the stacks aligned to the X-axis. It is possible, however, to center the columns vertically and create diagrams with a horizontal symmetry axis (like Wilkinson's symmetrical dot plots [36]). Further variations can align dot stacks to the Y-axis. This is especially useful when dealing with nominal data as it improves the layout of labels and creates an output similar to Cleveland's multiway dot plots [7]. In addition to layout and rendering variations, we can combine different visualization methods. For instance, Fig. 7b shows a symmetrical dot plot, overlaid by a box plot, whereas Fig. 3 adds strip charts to the X-axis. Such hybrid diagrams can help users classify and identify data in meaningful ways, providing more insights. Tukey's suspended rootograms plot data in relation to a known density distribution [31]. The same technique could be applied to the vertical positioning of the dot stacks to show deviations, but would require special care to adjust for areal distortions (see Section 4.4).



(a) Log_2 dot plot using $d_{\text{single}} = 795$ (Shneiderman only)

(b) Root dot plot using $s = 0.6$ and $d_{\text{single}} = 526$

(c) Log_2 dot plot using $d_{\text{single}} = 795$

(d) Linear dot plot using $d_{\text{single}} \approx 8.4$

(e) Logarithmic histogram

(f) Linear histogram

Fig. 6. Number of citations for papers by Ben Shneiderman, William S. Cleveland, Leland Wilkinson, and William E. Lorensen. Each dot represents one of 1,463 publications from these authors.

5 Examples

We demonstrate nonlinear dot plots for typical examples of real-world data. These contain frequencies of varying range. We include comparisons to histograms and linear dot plots to examine different characteristics.

5.1 Distribution of Air Temperature

Air temperature has a direct impact on our daily life, but it is also related to issues of global climate change. Therefore, we pick this application as our first example. The data was provided by the German meteorological service (Deutscher Wetterdienst)[2]. Our data set contains the aggregated mean of the maximum daily air temperature by weather station and month for the years 1961 to 1990. The stations are labeled using the identifiers supplied by the World Meteorological Organization (WMO). This data set provides a total of 9685 data samples for 875 weather stations worldwide; temperatures are in degrees Celsius.

Visualizing this data with the plots in Figs. 1 and 5, one can immediately see that the distribution is unimodal and has its maximum between 31 to 32 degrees. While mean temperatures down to around -30 degrees are still quite common, only single instances of temperatures at or below -31 degrees can be observed. Temperatures between 16 and 29 degrees are almost evenly distributed, forming a plateau in the linear dot plot. This plateau is also visible in the logarithmic histogram and the root dot plot, albeit less noticeable. The nonlinear plots outperform their linear counterparts when we focus on less dense areas of the plots. Figures 1 and 5c both show minimum and maximum values (outliers) clearly, but the dot plot also allows counting them, whereas the histogram would need a fine-grained vertical axis to do so, which would lead to overplotting. The extreme, strip chart-like visualization in Figure 5b only shows the outliers clearly but turns the dense regions into light-gray areas. Cross-referencing the temperature data with the WMO database yields additional geographic information. Picking data points in nonlinear dot plots for cross-referencing is very simple because of the one-to-one relationship between dots and data points and since the low-frequency points are rendered as quite large dots.

While the two nonlinear visualizations look similar at first glance, the dot plot provides more detail in dense areas. The histogram cannot provide such a view, as all its bins are all equal in width. Figure 1 also exhibits some gaps in the stacks (for instance near 7 °C), which contrast the tightly packed neighboring areas. This is due to the characteristics of the data source: the temperatures are rounded to a single decimal place and there are many data points with the same value. The value density in this area is too low to create high and narrow columns, but at the same time, it is too high to place a wider column at each decimal of a degree. Therefore, the layout algorithm cannot create a tightly packed field of uniform columns, which leads to visible gaps.

[2] https://cdc.dwd.de/catalogue/srv/en/home

To encode more information we colored the individual dots in Figs. 1 and 5a according to the four seasons. Seasons are different for Earth's northern and southern hemisphere. We compensated for that by introducing a phase-shift of six months for weather stations below the equator. Using this colorization with the nonlinear dot plot, it is immediately noticeable that higher temperatures tend to occur in summer, while the lower ones are predominantly measured in winter. That is no surprise; however, the tendency is not so obvious when looking at the linear dot plot. This effect can be explained by the unimodal sample distribution that has a lower value frequency at the outer edges and the bias toward outliers in Fig. 1. It would be possible to color the histograms analogously to the dot plots, by subdividing the bars. However, there is no appropriate color-coding for the logarithmic histogram because there is a conflict between linear (proportional) splitting of individual bars versus the overall logarithmic scaling. There is no such ambiguity with the colored dot plots, as the individual stacks are always linear, giving the users an impression of the distribution of sample categories within the stacks.

5.2 Citation Statistics

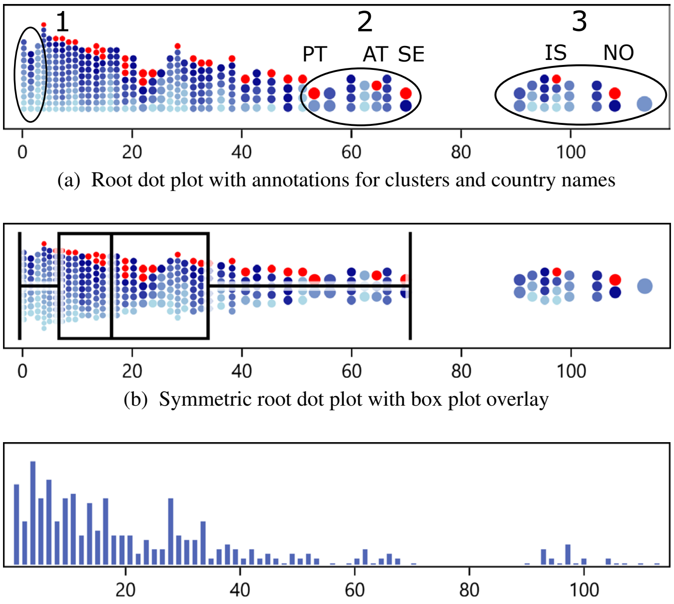
Our next example shows bibliometric data in the form of citation statistics. The h index [15] is a popular indicator of publication impact by an author, heavily aggregating data of all papers into a single number. In contrast, visualizing the complete citation data presents a challenge, as the number of citations may vary extremely per author and paper: there tend to be few publications with a big impact (i.e., many references) and many papers that hardly anyone notices (i.e., little to no references). Therefore, we obtain a high concentration in the frequency plot near zero and some outliers with many citations, forming a long tail. Since these outliers are the most relevant publications, the visualization should represent them accordingly. However, even the bulk of low-citation papers is interesting because it indicates publication productivity.

For illustration purposes, we use citation data of four well-known researchers (Shneiderman, Cleveland, Wilkinson, and Lorensen), obtained from *Google Scholar*[3] through the *Publish or Perish* software [14] (withouth any data cleansing). Figure 6 shows the results. In 6a, we plotted 1045 publications by Ben Shneiderman. His most cited work is "Designing the user interface: strategies for effective human-computer interaction" (14,309). There are six additional publications that are clearly distinguishable, but most of the other papers seem to have one thousand or fewer citations.

To compare his citation data to that of the other three authors, we add their data and use color-mapped dots. By the dominant dark blue color in the nonlinear dot plots in (b) or (c), we can see immediately that Shneiderman has the largest number of publications. Dots below about 1,000 citations become too small to distinguish individually, but their nonlinearly scaled column heights can still be perceived; therefore, we can obtain the approximate frequencies and compare them between authors. The much fewer papers with high citation counts are large and clearly visible. From these dot plots, we can see that all four authors have publications with 9,000 or more references. Bill Lorensen even has two papers with very high citation counts: "Marching cubes: A high resolution 3D surface construction algorithm" (13,495) and "Object-oriented modeling and design" (11,147). Leland Wilkinson's most cited work is "SYSTAT for Windows: statistics, graphics, data, getting started, version 5" (9,914). The red dot at 9,017 represents William Cleveland's "Robust locally weighted regression and smoothing scatterplots".

In contrast, linear plots (Figs. 6d and 6f) are not well suited for such high dynamic range data, as they cannot show any useful information about the important long tail. The logarithmic histogram in (e) renders the highly referenced publications as bars. This allows for a relatively accurate estimation of the citation counts, but does not provide any method of showing and comparing the authors.

[3] https://scholar.google.com



(a) Root dot plot with annotations for clusters and country names

(b) Symmetric root dot plot with box plot overlay

(c) Linear histogram

Fig. 7. Percentage of electricity produced from renewable energy sources versus total consumed electricity of 30 European countries from 2004 to 2014. There is a value for each country and year. Dot plots use a color scale to represent the value's year: '04 '05 '06 '07 '08 '09 '10 '11 '12 '13 '14.

5.3 Renewable Energy

Our third example combines dot plots with box plots to address the way in which electricity is produced. The European Commission provides the general public with access to statistical data through their website called eurostat[4]. It includes information on the amount and type of energy produced in each country of the European Union (EU) as well as Norway and Iceland. The data set "tsdcc330"[5] compares the generated renewable electricity with the total consumed amount on a yearly basis, from 2004 to 2014.

Visualizing this data set with dot plots and histograms, three groups of values stick out that can be interpreted by combining the use of plots with the tabular representation of the source data. As Fig. 7 and more specifically cluster 1 in (a) show, there is a relatively high data density near 0%. Temporal information from the dot plots shows that it must have been from countries that have only recently started producing electrical power from renewable sources in relevant amounts. This conclusion is evident because there are stacks around zero percent, but red dots (indicating the data set's last year) are no longer present.

A second cluster from Fig. 7a is less visible in the histogram (c). The red dots indicate that it is composed of values of approximately three countries. Assuming a general trend of increasing the amount of renewable energy (in order to meet climate protection goals), it seems probable that the country with the leftmost red dot has the lowest share of the cluster. A look at the tabular data confirms this hypothesis: a single dot is contributed by Portugal, while Austria and Sweden make up the remainder of the cluster.

Cluster 3 from Fig. 7a is well separated from most of the other values and is well discernible in all three plots. The overlaid box plot in (b) explicitly classifies the entire cluster as consisting of outliers. At first glance, there are two countries that mainly use renewable energy. A closer comparison and dot count show that each color is present exactly twice, further indicating that all values are only from those two countries. Cross-referencing with tabular data identifies them as Norway and Iceland.

[4] http://ec.europa.eu/eurostat/web/main/home

[5] http://ec.europa.eu/eurostat/tgm/download.do?tab=table&plugin=1&language=de&pcode=tsdcc330

Due to the lack of countability, the regular histogram was not as useful in arriving at these conclusions. The dot plots show individual data values that prompt for a more directed research into details.

6 Expert Review

To evaluate nonlinear dot plots, we conducted an expert review [28]. We recruited four experts from the field of visualization and computer graphics from our university. All of them were PhD students with several years of research experience. Our evaluation focused on the visualization itself, not the interactive tool used to generate the plots. Therefore, we presented each of the experts with static printouts on paper. These included four different kinds of visualizations: linear dot plots and histograms, nonlinear dot plots, and logarithmic histograms. Each review took approximately 30 minutes, during which an operator engaged the experts in a dialog (structured by a previously prepared sequence of questions) and took notes.

In the beginning of the review, the expert was introduced to the concept of dot plots and specifically to the nonlinear dot plots and stack sizes. Then, the expert was asked to work with all four plot types, each applied to three different examples of time-series data sets: (1) “FOL_acc_1_1” from the MobiFall[6] [32, 33] data set, which contains axis-dependent accelerations of people falling, measured with smartphone devices. The visualizations showed 812 samples. All plots were rendered in black, i.e., no additional color mapping was applied. (2) The energy data set discussed in Section 5.3. The dot plots were colored as in Figs. 7a and 7b, whereas the histograms remained black. (3) Temperature data showing minimum daily temperatures as in Fig. 1. In this case, the dot plots were also colored, whereas histograms remained black. The review session concluded with collecting the expert’s general feedback and remarks on the different data representation methods. Although we compare colored and black visualizations, we feel this is a suitable study setup, because it makes use of the most important advantage of dot plots (countability) and we would not want to compare techniques without their main features.

We first summarize our main observations made while the experts worked with the visualizations. For data set (1), all experts recognized a sharp peak near the number 10, but no one made a connection to the gravitational acceleration of approximately $9.81\,\mathrm{m\,s^{-2}}$. An expert with a pronounced background in statistics noticed the connection between dot plots and histograms, and found the analogy between dot stacks and narrow bins in high sample frequency areas. In low-density areas, however, the same visualizations diverged as the number of stacks and bars did not match. This was due to the bin limits that separated samples into a bar each, while nonlinear dot plots joined multiple values into low stacks. Everyone noticed the lack of perceivable bars in low-frequency areas in linear histograms (due to rasterization), while some were able to discern individual circles in linear dot plots.

Experts were first introduced to a color scale for the dots when presented with the energy data. The first objective was to find out how many countries were represented in the plot. This was to check whether they understood the meaning of a single dot and the color scale. One expert with much experience in statistics made the connection himself: the number of countries is equal to the number of red dots. The others needed additional explanations but eventually, everyone got a grasp of the concept and answered the next question correctly: There were no more countries without electricity from renewable energy sources in the last year. When asked whether the data presented outliers, all experts mentioned the cluster around 100 % and that at least in the last year, there were only two countries. The small difference between printed colors and the plot’s small size did not allow them to compare individual dots and come to the conclusion that the cluster always contained two dots per year.

The third group of visualizations came with estimation tasks. The experts should determine the 25, 50, and 75 percentiles in each plot separately. Everyone reported that it was much easier with linear representations than with nonlinear ones, as they had to compensate representational distortions. This is to be expected and in line with previous findings [1, 4, 8, 16, 30]. The estimation results reflected the expert’s statements, although the number of participants was too small to achieve statistical relevance. Two experts with a background in statistics were able to compensate the distortions more effectively and arrived at virtually the same results for each visualization.

After having gone through three data sets, we asked the experts which would be the first visualization (out of the four available) they would use on completely unknown data. Three preferred the linear histogram, mostly because they already knew it from previous work and were quite familiar with it. One preferred the nonlinear dot plot because outliers were better visible, with and without color mapping.

Finally, we asked for general feedback and remarks on the different data representation methods. The consensus was that dot plots are aesthetically pleasing, especially when combined with color mapping. The nonlinear representation allows for identifying and counting outliers. However, there is the drawback of nonlinear projections, as they introduce distortions for which the observer has to compensate. The experts were intrigued by the possibilities of color mapping and counting individual data samples, and mentioned that it allowed for the extraction of more details from the visualizations.

7 Conclusion and Future Work

We have introduced nonlinear dot plots—a new type of diagram to show the distribution of data. Nonlinear dot plots combine the advantages of conventional dot plots and nonlinear scaling known from log-scale histograms. Therefore, nonlinear dot plots are well prepared to visualize countable data samples for data sets with a large range of frequencies. With our new type of plot, we are able to visualize data distributions with well above a thousand individual values; still, we can easily discover outliers because the single data points are drawn as large dots in areas of few data points. In our experience, linear dot plots work just as well if the data sets are small (less than 1,000 values) and the value frequency stays within a small range.

Computing the layout becomes more difficult once we generalize conventional dot plots to nonlinear ones. Therefore, we have presented a new two-way sweep algorithm that produces intermediate dot plots by traversing the data domain in upward and downward directions; these intermediate layouts are combined to obtain the final plot. We have also addressed the problem of aliasing moiré effects by devising a specific low-pass filtering method that performs controlled vertical blurring along inner parts of the dot columns.

Nonlinear dot plots have a wide range of possible uses, similar to other frequency plots. We have also demonstrated how they can be combined with other visualization elements, including color-coding and hybrid overlays. Some real-world examples and an expert review have demonstrated the utility and characteristics of nonlinear dot plots. It should be noted that the idea of nonlinearly scaling dot sizes is more fundamental than the two-way sweep algorithm we presented. In fact, there might be future improvements for the actual layout algorithm. For example, iterative approaches like simulated annealing or genetic algorithms might also prove to give satisfactory layout results. Just as node-link diagrams of graphs can be laid out by force-directed methods [12, 19, 20] with randomization elements, the creation of columns could be implemented by shifting dots, merging, and splitting columns.

We currently only use color to group the dots in a column. Further development could result in other techniques for such comparative dot plots. As another element of future work, we plan to conduct a controlled user study with eye-tracking to obtain objective measures for the differences between nonlinear dot plots and other frequency plots.

Acknowledgments

We would like to thank the German Research Foundation (DFG) for financial support within project B01 of SFB/Transregio 161. Special thanks to Rudolf Netzel, Ayush Kumar, Markus Huber, and Qi Han for their participation in the expert review as well as Tanja Blascheck and Christoph Schulz for their feedback.

[6] http://www.bmi.teicrete.gr

REFERENCES

[1] C. Arbesser, F. Spechtenhauser, T. Mühlbacher, and H. Piringer. Visplause: Visual data quality assessment of many time series using plausibility checks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):641–650, 2017.

[2] P. Armitage, G. Berry, and J. N. S. Matthews. *Statistical Methods in Medical Research*. John Wiley & Sons, 2008.

[3] A. Bakker. *Design research in statistics education: On symbolizing and computer tools*. PhD thesis, Utrecht University, 2004.

[4] R. Borgo, J. Dearden, and M. W. Jones. Order of magnitude markers: An empirical study on large magnitude number detection. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2261–2270, 2014.

[5] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis*. Wadsworth, Belmont, CA, 1983.

[6] W. S. Cleveland. *The Elements of Graphing Data*. Wadsworth, Monterey, CA, 1985.

[7] W. S. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[8] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.

[9] Y. Copin, N. Cretton, and E. Emsellem. Axisymmetric dynamical models for SAURON and OASIS observations of NGC 3377. *Astronomy & Astrophysics*, 415(3):889–903, 2004.

[10] T. N. Dang, L. Wilkinson, and A. Anand. Stacking graphic elements to avoid over-plotting. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1044–1052, 2010.

[11] T. G. Fechner. *Elemente der Psychophysik*. Breitkopf & Härtel, Leipzig, 1860.

[12] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[13] L. Glass et al. Moire effect from random dots. *Nature*, 223(5206):578–580, 1969.

[14] A.-W. Harzing. Publish or perish. http://www.harzing.com/pop.htm, 2007. Accessed: 2017-08-01.

[15] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46):16569–16572, 2005.

[16] M. Hlawatsch, F. Sadlo, M. Burch, and D. Weiskopf. Scale-stack bar charts. *Computer Graphics Forum*, 32(3):181–190, 2013.

[17] P. Isenberg, A. Bezerianos, P. Dragicevic, and J.-D. Fekete. A study on dual-scale data charts. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2469–2478, 2011.

[18] W. S. Jevons. *Investigations in Currency and Finance*. Macmillan and Co., 1884.

[19] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

[20] S. G. Kobourov. Spring embedders and force directed graph drawing algorithms. *arXiv preprint arXiv:1201.3011*, 2012.

[21] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, 1994.

[22] S. Padmi and I. G. A. Russasmita. From sample to population: The use of dot plot to support the emergence of informal inferential reasoning. In *The Third South East Asia Design/Development Research International Conference*, pages 314–324, 2015.

[23] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355–368, 1987.

[24] M. Räisänen, M. Virkkunen, M. Huttunen, B. Furman, and J. Kärkkäinen. Increased urinary excretion of bufotenin by violent offenders with paranoid symptoms and family violence. *The Lancet*, 324(8404):700–701, 1984.

[25] P. D. Sasieni and P. Royston. Dotplots. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 45(2):219–234, 1996.

[26] D. W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

[27] A. Steinbach and K. Y. Wong. Moire patterns in scanned halftone pictures. *Journal of the Optical Society of America*, 72(9):1190–1198, 1982.

[28] M. Tory and T. Möller. Evaluating visualizations: Do expert reviews work? *IEEE Computer Graphics and Applications*, 25(5):8–11, 2005.

[29] M. Tory, C. Swindells, and R. Dreezer. Comparing dot and landscape spatializations for visual memory differences. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1033–1040, 2009.

[30] E. R. Tufte and P. Graves-Morris. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983.

[31] J. W. Tukey. Some graphic and semigraphic displays. In T. Bancroft, editor, *Statistical Papers in Honor of George W. Snedecor*, pages 293–316. Iowa State University Press, 1972.

[32] G. Vavoulas, M. Pediaditis, C. Chatzaki, E. G. Spanakis, and M. Tsiknakis. The MobiFall dataset: Fall detection and classification with a smartphone. In Information Resources Management Association, editor, *Artificial Intelligence: Concepts, Methodologies, Tools, and Applications*, pages 1218–1231. IGI Global, 2017.

[33] G. Vavoulas, M. Pediaditis, E. G. Spanakis, and M. Tsiknakis. The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones. In *Proceedings of the IEEE International Conference on Bioinformatics and Bioengineering*, pages 1–4, 2013.

[34] M. P. Wand. Data-based choice of histogram bin width. *The American Statistician*, 51(1):59–64, 1997.

[35] H. Wang and M. Song. Ckmeans.1d.dp: Optimal *k*-means clustering in one dimension by dynamic programming. *The R Journal*, 3(2):29–33, 2011.

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

[37] J. B. Zimmerman, S. M. Pizer, E. V. Staab, J. R. Perry, W. McCartney, and B. C. Brenton. An evaluation of the effectiveness of adaptive histogram equalization for contrast enhancement. *IEEE Transactions on Medical Imaging*, 7(4):304–312, 1988.

[38] K. Zuiderveld. Contrast limited adaptive histogram equalization. In P. S. Heckbert, editor, *Graphics Gems IV*, pages 474–485. AP Professional (Academic Press), Boston, MA, 1994.