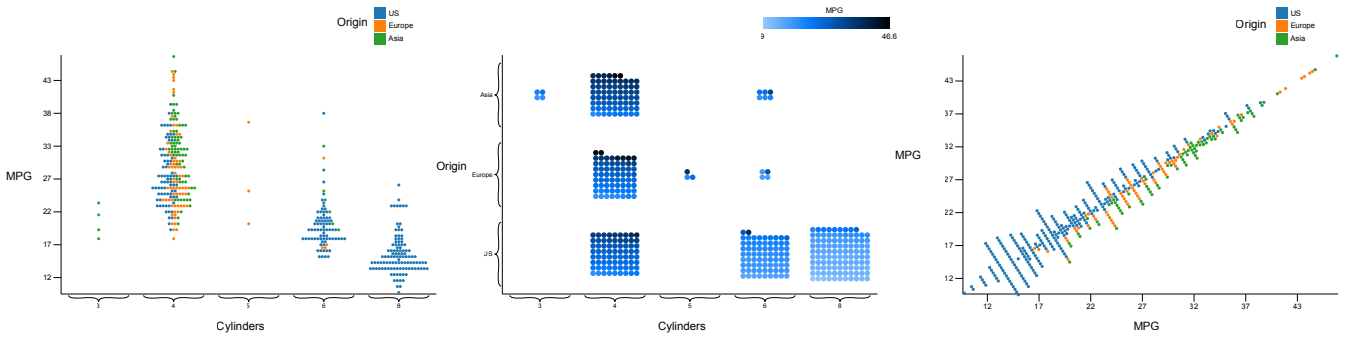
Gatherplots: Extended Scatterplots for Categorical Data

1st Author Name
Affiliation
Address
e-mail address
Optional phone number

2nd Author Name
Affiliation
Address
e-mail address
Optional phone number

3rd Author Name
Affiliation
Address
e-mail address
Optional phone number



(a) Cylinders vs. MPG (b) Cylinders vs. Origin (c) MPG vs. MPG

Figure 1. Multiple gatherplots show dataset related to cars, which created overplotting in scatterplots. The gatherplot in (a) shows Cylinders vs. MPG, which are a categorical and an continuous variable. The gatherplots show the overall distribution of MPG values of cars with different cylinders. The brackets on the X-axis are used to indicate that the space within brackets represent same value in the data. The gatherplot in (b) shows Cylinders vs. origin, which is a categorical vs. categorical variable case. The gatherplots partition the graphical axes into intervals and stacks points into groups for each interval. In (c), both X-axis and Y-axis show same continuous variable, MPG. In scatterplots, all these cases create an overplotting, which results in lines (a, c) or dots (b)

ABSTRACT

Scatterplots have been used for exploration of multi-dimensional dataset, especially in the form of scatterplot matrices (SPLOM). However there is an overplotting in SPLOM, when categorical variables are mapped to one or two axes or the same continuous variables are used for both axes. To ameliorate this, we propose gatherplots, an extension of scatterplots to handle these cases better. In gatherplots every data point that maps to the same position coalesces to form a stacked entity, thereby making it easier to see the overview of data groupings. The size and aspect ratio of data points can also be changed dynamically to make it easier to compare the composition of different groups. In the case of a categorical variable vs. a categorical variable, we propose a heuristic to decide bin sizes for optimal space usage. The gatherplots and the resulting gatherplot matrices (GPLOM) show enhanced utilization of spaces to show the overall distribution. Our evaluation shows that gatherplots enable users from the general public to judge the relative portion of subgroups more quickly more correctly than when using conventional scatterplots with jittering.

Author Keywords

Scatterplots; overplotting; scatterplot matrices (SPLOM)

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Scatterplots—one of the most widely used types of statistical graphics [5, 11, 31]—are commonly used to visualize two continuous variables using visual marks mapped to a two-dimensional Cartesian space, where the color, size, and shape of the marks can represent additional dimensions. Recently it has been used for the exploration of multi-dimensional dataset. The scatterplot matrices (SPLOM), where all the possible combination of axes are iterated in table form, is frequently used to show overview of multi-dimensional datasets.

However, realistic multidimensional datasets often contain categorical variables, such as nominal variables or discrete data dimensions with small domain. Even if a dataset is composed of entirely continuous variables, the SPLOM shows an overplotting along the diagonal axis, where the same variables are assigned to the both axes. As can be seen in Figure 2, there are three situations, when the overplotting is inevitable.

- When nominal variables have been used for both axes, the scatterplots results in dots pattern such as figure 2(d).

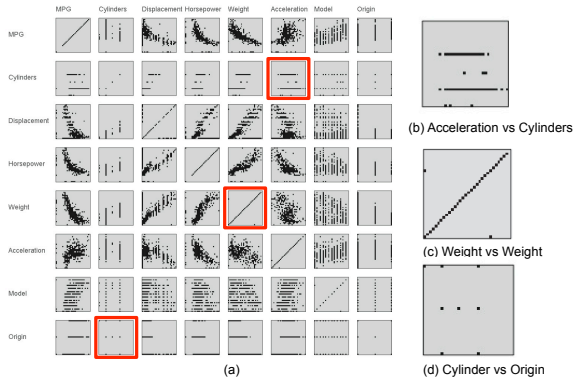


Figure 2. Limitations of scatterplots for managing discrete variables. (a) The scatterplot matrix for 7-dimensional car dataset. (b) The acceleration vs. cylinders scatterplot looks like a horizontal line because the cylinders variable works as a categorical variable, which causes overplotting. (c) The weight vs. weight scatterplot shows diagonal line because the values are mapped one single line. This problem is shown at all diagonal subfigures in the scatterplot matrices. (d) The cylinders vs. origin scatterplot shows dotted patterns because discrete dimensions have been mapped onto both the X and Y axis.

- When a nominal variable and an ordinal variable have been used together, the scatterplots shows overdrawing patterns with horizontal lines or vertical lines such as figure 2(b).

- When a same ordinal variable has been used for both axes, the resulting scatterplots shows a diagonal line, such as figure 2(c). This pattern is shown along the diagonal axis of the SPLOM.

Several approaches have been proposed to address this problem [10], the most prominent transparency, jittering, and clustering techniques. The first of these, changing transparency, does not so much address the problem as sidestep it by making the visual marks semi-transparent so that an accumulation of overlapping points in the same are still visible. However, this will not scale well for large datasets, and also causes blending issues if color is used to encode additional variables. Jittering perturbs visual marks using a random displacement [29] so that no mark falls on the exact same screen location as any other mark, but this approach is still prone to overplotting for large datasets. Jittering also introduces uncertainty in the data that is not aptly communicated by the scatterplot since marks will no longer be placed at their true location on the Cartesian space. Clustering, on the other hand, attempts to organize overlapping marks into visual groups that summarize the distribution [14, 22, 20]. However these clustering create histogram or kernel density estimation(KDE), which does not follow the visual grammar of scatterplots. Especially they lose the concept of the individual object, which can be problematic in the filtering or search tasks.

In this paper, we propose the *gatherplots* to overcome these limitations. Gatherplots generalizes the linear mapping used by scatterplots. And it partitions the graphical axis into segments based on the data dimension. Then it organizes points into *stacked groups* for each segment that avoids overplotting. This means that the gatherplots relaxes the continuous spatial mapping traditionally used for a graphical axis; instead, each discrete segment occupies a certain amount of screen space that is all defined to map to the exact same data value. This is also visually communicated using graphical brackets on the axis that show the value for each segment (Figure 1(b)).

The contributions of our paper are the following: (1) the gatherplots which extends scatterplots to categorical variables, maintaining individual objects; (2) the heuristic way to set the optimal dot size when there are a continuous variable and a categorical variable; and (3) results from a crowdsourced user study on the effectiveness of different modes of gatherplots. In the remainder of this paper, we first review the literature on scatterplots and overplotting. We then present gatherplots and discusse various design choices. This is followed by our crowdsourced evaluation. We close with implementation notes, conclusions, and our future plans.

BACKGROUND

Our goal with gatherplots is to generalize scatterplots to a representation that maintains its simplicity and familiarity while eliminating overplotting. With this in mind, below we review prior art that generalizes scatterplots for mitigating overplotting. We also discuss related visualization techniques specifically designed for nominal variables.

Characterizing Overplotting

While there are many ways to categorize visualization techniques, a particularly useful classification for our purposes is one introduced by Fekete and Plaisant [13], which splits visualization into two types:

- **Overlapping visualizations:** These techniques enforce no layout restrictions on visual marks, which may lead to them overlapping on the display and causing occlusion. Examples include scatterplots, node-link diagrams, and parallel coordinates.

- **Space-filling visualizations:** A visualization that restricts layout to fill the available space and to avoid overlap. Examples include treemaps, adjacency matrices, and choropleth maps.

Fekete and Plaisant [13] investigated the overplotting phenomenon for a 2D scatterplot, and found that it has a significant impact as datasets grow. The problem stems from the fact even with two continuous variables that do not share any coordinate pairs, the size ratio between the visual marks and the display remains more or less constant. Furthermore, most datasets are not uniformly distributed. This all means that overplotting is bound to happen for realistic datasets.

Ellis and Dix [10] survey the literature and derive a general approach to reduce clutter. According to their treatment, there are three ways to reduce clutter in a visualization: by changing the visual appearance, through space distortion, or by presenting the data over time. Some trivial but impractical mechanisms they list include decreasing mark size, increasing display space, or animating the data. Below we review more practical approaches based on appearance and distortion.

Appearance-based Methods

Practical appearance-based approaches to mitigate overplotting include transparency, sampling, kernel density estimation (KDE), and aggregation. Transparency changes the opacity of the visual marks, and has been shown to convey overlap for up to five occurrences [34]. However, there is still an upper limit for how much overlap is perceptible to the user, and the blending caused by overlapping marks of different colors makes identifying specific colors difficult. Sampling uses stochastic methods to statistically reduce the data size for visualization [9]. This may reduce the amount of overplotting, but since the sampling must be random, it can never reliably eliminate it. Furthermore one of fundamental strength in scatterplots is its ability to show outliers effectively. However with sampling the outliers will be removed from the visual space.

KDE [27] and other binned aggregation methods [12, 14, 22, **?**] replace a cluster of marks with a single entity that has a distinct visual representation. Splatterplots [22] overcome this by overlaying individual marks side-by-side with the aggregated entities, using marks to show outliers and aggregated entities to show the general trends. This remedies the problem with KDE and sampling, for it maintains the outliers while showing overviews. However, as pointed by authors, even with only few aggregated entities, the resulting color-blended image becomes visually complex and challenging to read and understand. Also these technique can not be applied for the cases when there are categorical variables, for KDE method tend to smooth out the granularity of datasets. Generalized plot matrix(GPLOM) is proposed to solve this problem. This pioneering work solves the overplotting by adopting non-homogeneous plots into matrix. GPLOM uses a histogram for categorical vs. continuous variables and a treemap for categorical vs. categorical variables. While effective in providing overview, it loses logical compatibility with scatterplots, for scatterplots operate on the principle of object identity, meaning that each visual mark is supposed to represent a single entity.

Distortion-based Methods

Distortion-based techniques provides advantage that it keeps individual object. The canonical distortion technique is jittering, where a random displacement is used to subtly modify the exact screen space position of a data point. This has the effect of spreading data points apart so that they are easier to distinguish. However, most naïve jittering mechanisms apply the displacement indiscriminately to all data points, regardless of whether they are overlapping or not. This has the drawback of distorting all points away from their true location on the visual canvas, and still does not completely eliminate overplotting.

Bezerianos et al. [2] use a more structured approach to displacement, where overlapping marks are organized onto the perimeter of a circle. The circle is grown to a radius where all marks fit, which means that its size is also an indication of the number of participating points. However, this mechanism still introduces uncertainty in the spatial mapping, and it is also not clear how well it scales for very dense data. Nevertheless, it is a good example of how deterministic displacement can be used to great effect for eliminating overplotting.

Trutschl et al. [29] propose a deterministic displacement ("smart jittering") that adds meaning to the location of jittering based on clustering results. Similarly, Shneiderman et al. [26] propose a related structured displacement approach called *hieraxes*, which combines hierarchical browsing with two-dimensional scatterplots. In hieraxes, a two-dimensional visual space is subdivided into rectangular segments for different categories in the data, and points are then coalesced into stacked groups inside the different segments. This inspiration laid the foundation for our extensions that will refine the layouts and design. Especially according to Haroz et al. [15], grouping nodes of similar visual feature helps performing tasks such as finding outliers, number of different classes and so on.

Visualizing Categorical Variables

While we have already ascertained that scatterplots are not optimal for categorical variables, there exists a multitude of visualization techniques that are [1, 18, 21]. Simplest among them are histograms, which allows for visualizing the item count for each categorical value [28], but much more complex representation are possible. While hieraxes, histograms or treemap are effective in dealing with nominal variables, it is difficult to extends these to continuous variable vs categorical variables. One way to extend is applying binning to continuous variables to create groups of values. However the optimal number of bin depends on statistical characteristics of data and required task. Dot plots by Wilkinson [32] shows continuous univariate variable without overplotting by stacking nodes within dot size. Dang et al. [8] extended this to scatterplots by stacking nodes whose values are similar in 3D visual space. These pioneering work provided theoretical background for the determination of the optimal bin size for the gatherplots.

One particular usage for visualizing categorical data that is of practical interest is for making inferences based on statistical and probabilistic data. Cosmides and Toody [7] used frequency grids as discrete countable objects, and Micallef et al. [24] extend this with six different area-proportional representations of nominal data organized into different classes. Huron et al. [19] suggested sedimentation as metaphor where individual objects coming from data stream gradually transforms into aggregated areas or strata.

DESIGN OF GATHERPLOTS

Here we explain the design of gatherplots and rationale behind each design choices. Gatherplots alleviate overplotting, focusing on optimal layouts of gathered entities, graphical representations of chart elements, and novel interactions. Especially gatherplots deal with overplotting when there are one or two nominal variables in the dataset. There are many open design possibilities for aspect ratio, layout, and item shapes. We discuss these design parameters in the treatment below.

Categorical Variable vs. Categorical Variable

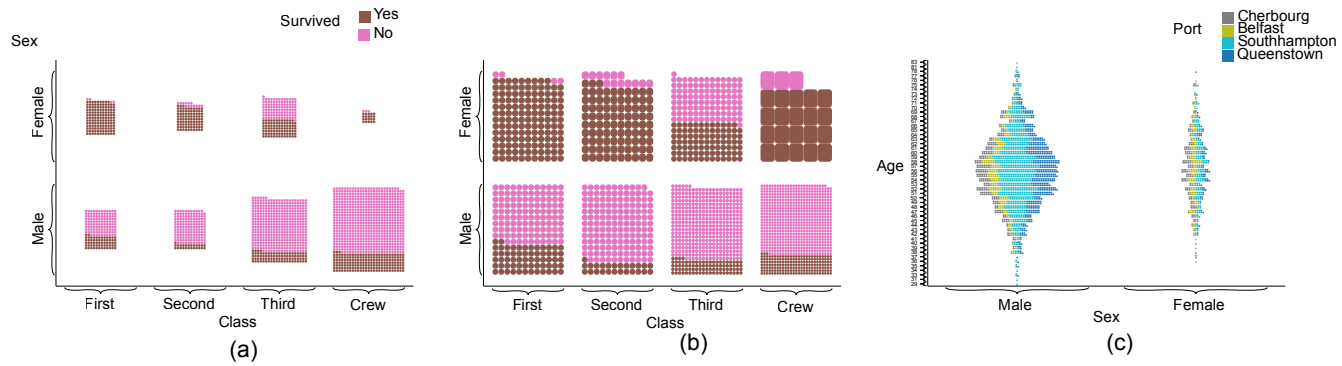


Figure 3. Main layout modes for gatherplots: (a) absolute mode with constant aspect ratio, which arranges items following the aspect ratio of given area; (b) relative mode of (a). The rate of survivors in each male passenger class is not each to compare. Figure (c) shows the streamgraph mode, where each cluster maintains the number of element in the shorter edge, making it easier to see the distribution of the subgroups along the Y axis.

Previous works such as hieraxes [26] or frequency grids [24, 7] organize entities into *stacked groups* according to a discrete variable to eliminate overplotting. Gatherplots follows this approach and applies the gathering of nodes with similar visual properties. According to Haroz and Whitney [15], a cluttered randomly organized arrangement lowers the search speed for the target. However there are many design possibilities organizing visual representation depending on the context, especially on the size distribution of each groups, the aspect ratio of assigned space, and the task at hand. As a result, we derive the following three layout modes (examples in Figure 3):

- **Absolute mode.** Here stacked groups are sized to follow the aspect-ratio of the assigned region. The node size of the items are determined by the maximum length dots which can fill the assigned region without overlapping. This means with the same assigned space, the groups with the maximum number of members determines the overall size of the nodes (Figure 3(a)).

- **Relative mode.** In this mode, the node size and aspect ratio is adapted so that every stacked group has equal dimensions. This is a special mode to make it easier to investigate ratios when the user is interested in the relative distributions of subgroups rather than the absolute number of members. Items also change their shape from a circle (absolute mode) to a rounded rectangle (Figure 3(b)).

- **Streamgraph mode.** Here stacked groups are reorganized so that the maintain the same number of elements in their shorter edge. This mode is used for regions where the ratio of width and height are drastically different (in our prototype implementation, we use a heuristic value of 3 for aspect ratio to be a threshold for activating this mode). This means there are usually many times more groups in the axis in parallel with shorter edges. A good example can be when we want to see the distribution of population with regards to the gender variable and the age variable. The resulting graphic resembles ThemeRiver [17] as the number of entities increase (Figure 3(c)).

The choice between absolute mode and streamgraph mode happens automatically based on the aspect ratio of assigned space and without user intervention. Therefore only interactive option is required to toggle between absolute mode and relative mode. Our intuition is that the absolute mode should be good enough for most of the time, and when very specific tasks are required, the user can switch to the relative mode.

However, gatherplots involve many more possibilities beyond than these layout functions. Below follows our treatment of these design possibilities and our rationale for our decisions.

Area vs. Length Oriented Layout

Maintaining the aspect ratio of all stacked groups means that the size of the group is represented by its area. The length of the group is only used in special cases when the aspect ratio is very high or low. According to Cleveland and McGill [6], length is far more effective than the area for graphical perception. However, Figure 4 shows the three problems associated with layout to enable length-based size comparison. In this view, the items are stacked along the vertical axis to make the size comparison along the horizontal axis easier. The width of rectangle is all set to be equal to so that the length can represent the size of subgroups. However, they show drastically different shape of line vs. rectangle, which may cause users to lose concept of equality. Furthermore, to make length-based comparison easier, the stacking should be aligned to one side of the available space: left, right, top, or bottom.

In this case, the bottom is selected to make it easier to compare along the X axis. However, this creates additional two problems. The first problem is that the center of mass of each stacked group is too different that the concept of belonging to the same value can be misleading. The second problem is that choosing alignment direction is arbitrary and depends on the task. For example, in this view it is more difficult to compare along the Y axis. In this sense, this layout is biased to the X axis, while sacrificing the performance along the Y axis. For this reason, the most general choice is to use center alignment with aspect ratio resembling the assigned range to avoid bias.

Uniform vs. Variable Area Allocation

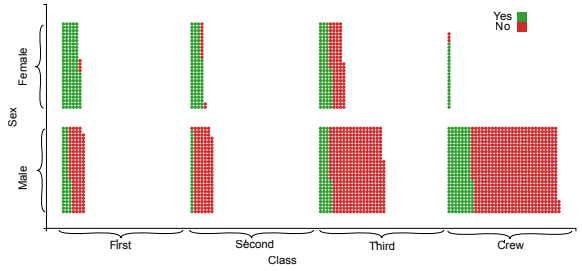


Figure 4. Stacked group layouts for gathering. This layout supports comparison group sizes. Because the height of stacked groups is all fixed to the same value, comparing the length yields the size.

In gatherplots, we assign uniform range to different values of overlappable variables. For some cases, assigning variable area can make sense and create interesting visualizations. As a simple example, we can argue that assigning the range of output for gather transfer function to be proportional to the numbers of items that belong that value uses the space most efficiently. This will result in the following layout shown in Figure 5, which can be reduced to a mosaic plot [16] or treemaps [1]. The use of variable area allocation generally makes better use of given display space. But in gatherplots we choose uniform area allocation because the scheme of scatterplot assumes uniform space among entities.

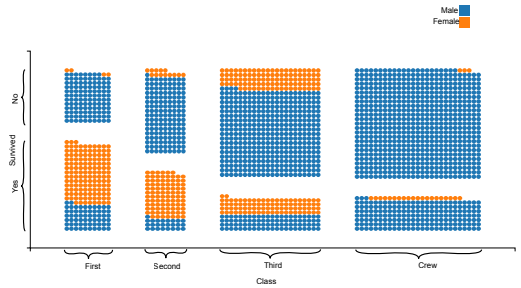


Figure 5. Variable area for a gatherplot. The chart uses space efficiently. Items which belong to same place should belong to the same value. However, as can be shown in the Y axis tick marker, this results in a violation of the scatterplot concept.

Role of Relative Mode

Since gathering assigns a discrete noncontinuous range to each graphical axis, each stacked group can be grown to fill all available space. This relative mode is useful for two specific tasks:

- Getting a relative percentage of the subgroups in the group (Figure 3). Because groups of different size is normalized to the same size, any comparison in area results in a relative comparison. This can aid the statistical bayesian reasoning [24].
- Finding the distribution of outliers. When there are many items on the screen for absolute mode, all node sizes must be reduced. This can make outliers hard to locate. When relative mode is used, the outliers are expanded to fill the assigned space, making it easier to notice it.

Continuous ordinal Variable vs. Categorical Variable

To create organized stacking of values over continuous ordinal variables, binning is applied in gatherplots. The bin size is important because it determines the spatial accuracy and viewing affordance. Pioneering work by Wilkinson [32] provides theoretical background for the optimal size of bins. Wilkinson proposed the $.25n^{-1/2}$ as the optimal dot size for dot plots. This is based on the assumption of normal distribution of data and aspect ratio of chart to be about 5.

However gather plot requires two different assumption. First the maximum width of dot plot is limited by space allocation by nominal variables. Second the dot size or bin size is determined by global maximum in dataset, which may not be in same nominal. For example, when we assume the gender Vs. height, and the male group is more packed than female group, the bin size of female group is determined by the maximum bin size of male group. To get the optimal bin size under these conditions, we propose following algorithm.

1. We begin with the number of bins to be 1.
2. The dot size or bin size becomes assigned height divided by number of bins.
3. Given the bins, find the maximum number of members in each bin for the entire dataset.
4. Calculate the required width by multiplying maximum number of members by dot size.
5. If the required width exceeds assigned width, increase the number of bins by 1 and go to step 2. Otherwise, use the dot size for the optimal bin size.

In practice, there is heuristic value of maximum dot size, where the dot size larger than that does not increase the viewing comfortability. And initial value of number of bins can be assigned height divided by maximum dot size to reduce computation times.

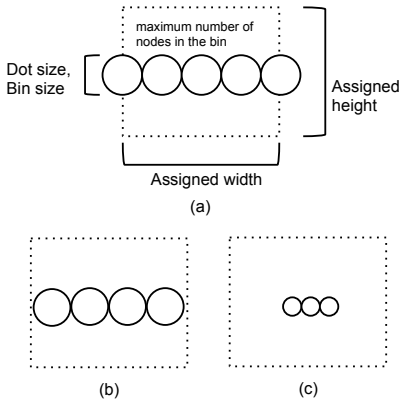


Figure 6. This diagram shows the steps for getting optimal bin size. In (a) the dot size is too large. The required width, which are the maximum number of nodes in the bin multiplied by dot size, exceeds the assigned width. In (b) the dot size is optimal, meaning the space is utilized fully. In (c) the dot size is too small, which results in the waste in the visual space.

Also the case of scatterplots with the same continuous variables can be treated as the special case of the continuous vs

nominal variable, where nominal variable is whole. The gatherplots are rotated to maintain integrity with scatterplots as in Figure 1 (c).

Visual Design and Interactions for gatherplots

Here we discuss the visual design choices such as shape and tick marks. Also we explain the novel interaction for the gatherplots called axis folding.

Continuous Color Dimension

The gatherplot sorts items according to a data property, such as a variable also assigned for coloring items. This removes the scattered color patterns in the stacked groups that is common in other techniques such as Gridl [26]. This is also particularly useful for continuous color scales, making the variation of colors are easier to perceive (Figure 7).

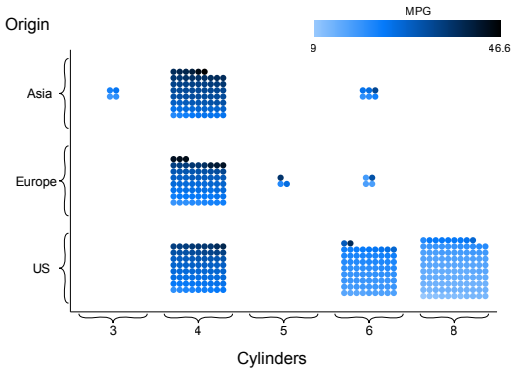


Figure 7. Continuous color scale used in a gatherplot. The X axis is the number of cylinders and and the Y axis is the origin of cars. The color scale is MPG.

Shape for Items

Scatterplots typically use a small circle or dot as a visual representation for items, but many variations exist that use glyph shapes to convey multidimensional variables[23, 30, 3, 5, 4]. However, in the relative mode, sometimes the aspect ratio of nodes changes according to the aspect ratio of box assigned to that value. Also, as gathering changes the size of nodes to fit in one cluster, sometimes node size becomes too small, or too large compared to other nodes. This results in several unique design considerations for item shapes. After trying various design alternatives, we recommend using a rectangle with constant rounded edge without using stroke lines. Using constant rounded edge allows the nodes to be circular when the node is small, as in Figure 3(b), and a rectangle to show the degree of stretching, as shown in Figure 3(b). Figure 8 shows some previous trials with various shapes.

Design of Tick Marks

The single line type tick marks for scatterplots are not appropriate for gatherplots. Because we are representing a range rather than a single point, a range tick marker will be better. Without this visual representation, when the user is confronted with a number, it can be confusing to determine whether adjacent nodes with different offset has same value or not. After considering a few visual representation, we recommend a bracket type marker for this purpose. Figure 9

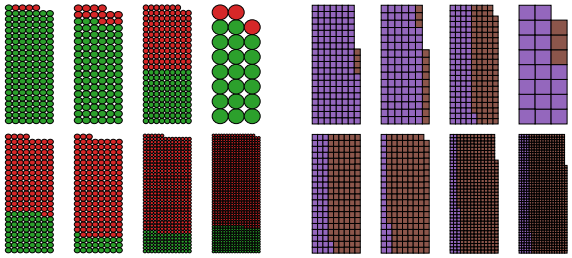


Figure 8. Stroke line problems where the circle consumes ample white space between adjacent nodes, which contributes to clutter as it grows. The rectangle does not have space between nodes, however, it must have a stroke border to show stretching. But this borderline creates problems when the items are very small.

shows various types of markers for range representation. The bracket is optimal in that it uses less ink and creates less density with adjacent ticks.

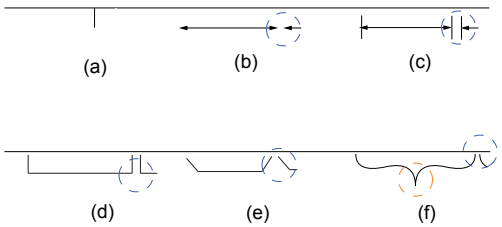


Figure 9. Various tick marks types. The blue dotted region represents the area between adjacent tick marks. (a) is a typical line type tick mark for the scatterplots. (b) lacks guide lines, which will make anchoring easier. (c) creates a packed region between adjacent marks. (d) uses less crowded region in this region, but (e) is the least crowded. (f) is the final recommendation, with the data label in the orange region.

Applications for Continuous Variables

Gatherplots can be used to mitigate overplotting caused by continuous variables as well. Figure 10 (a) shows how gatherplots handle the overplotting caused by continuous variables. The plot is using relative mode with two random variables. The relative mode makes it easier to identify the outliers and the distribution of outliers.

One limitation of gatherplots is that it requires binning to manage a continuous variable, yet binning creates arbitrary boundaries. In this sense, gatherplots can be misleading. However, combining gatherplots with scatterplots makes this problem less severe.

Axis Folding Interaction

As an exploration tool for real-world dataset, it is crucial to have means to filter unwanted data. To aid this process with gather transform, we provide an optional mechanism to go back to the original continuous linear scale function. We allow each axis tick have an interactive control to be filtered out (minimize) or focused (maximized). This is called *axis folding*, because it can be explained mentally by a folding paper. When minimized or folded, the visualization space is shrunk

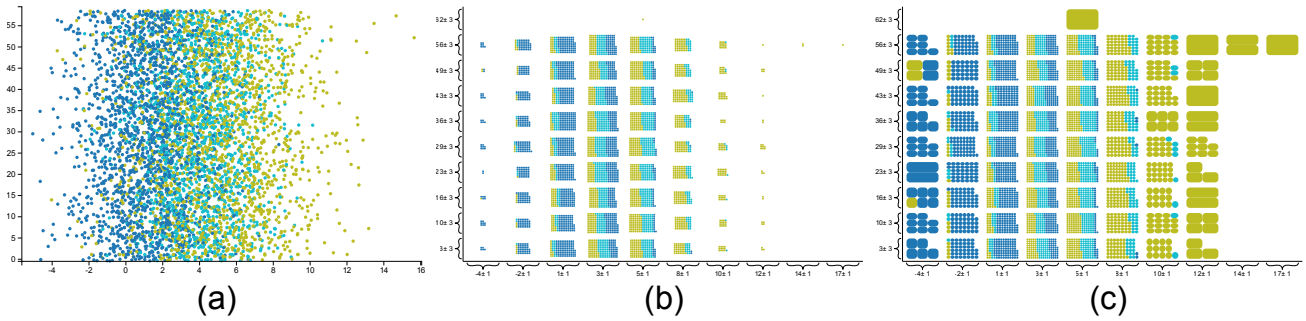


Figure 10. Using gatherplots to manage overplotting. (a) shows a scatterplot with 5,000 random numbers with severe overplotting in the center area. In (b), gathering is applied to create a more organized view. However, the gathering resizes the items so small that it becomes difficult to detect outliers. (c) shows relative mode, where the outliers are enlarged. This makes identifying the distribution of sparse regions easier.

by applying linear scales instead of nonlinear gather scales. This results in overplotting as if a scatterplot was used for that axis. A maximization is simply folding all other values except the value of the interest in order to assign maximum visual space to that value. Figure 11 shows the axis folding applied to third class adult passengers in the Titanic dataset.

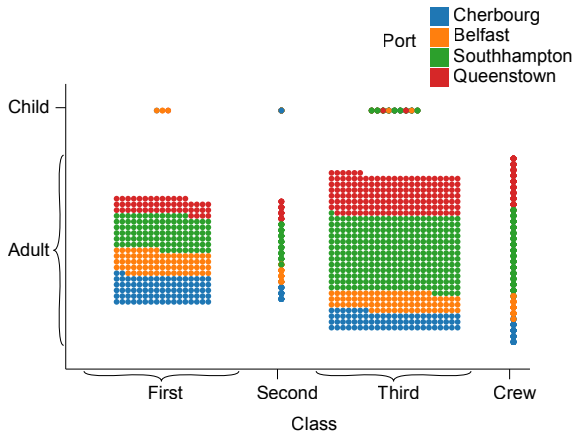


Figure 11. Survivors of the Titanic using gatherplots. The X axis is class of passengers, where second class passengers and crew are minimized. The Y axis is age, where the adult value is maximized. This view makes it easy to compare first class adults and third class adults. Note that even in the minimized state, we can get an overview about the second class and crew by the color line, which communicates the underlying distribution. This is due to the sorting over the color dimension.

EVALUATION OF GATHERPLOTS

The purpose of this study is to examine the effectiveness of gatherplots especially to see how different modes of gatherplots influence certain types of tasks for the crowdsourced workers. We have conducted the study for one of the particular cases, categorical variable vs. categorical variable. Crowdsourcing platforms have been widely used and have shown to be reliable platforms for evaluation studies [25, 33]. Therefore, we conducted our experiment on Amazon Mechanical Turk [1] .

Experiment Design

Gatherplots was developed to overcome limitations of conventional scatterplots. Jittered scatterplots were selected as baseline condition, as it is widely accepted standard technique maintaining same consistency with scatterplots. We also wanted to measure how different modes of gatherplots were effective. Therefore we designed the experiment to have four conditions such as scatterplots with jittering (jitter), gatherplots with absolute mode (absolute), gatherplots with relative mode (relative), and gatherplots with one check button to switch between absolute and relative mode (both). We adopted between-subject design to eliminate learning effect by experiencing other modes. The exact test environment is available for review [2]. Note the questions for each conditions were generated randomly.

Participants

A total of 240 participants (103 female) completed our survey. Because some questions asked a concept of absolute numbers and probability, we limited demographic to be United States to remove the influence of language. Also to ensure the quality of the workers, qualification of workers were the approval rate of more than 0.95 with number of hits approved to be more than 1,000. Only three of 240 participants did not use English as their first language. 119 people had more than bachelor's degree, with 42 people having high school degree. We filtered random clickers, if the time to complete one of questions was shorter than a reasonable time, 5 seconds. Eventually, we have a total of 211 participants.

Task

As scatterplots can support various types of tasks, it is difficult to come up with a representative task. After reviewing tasks for categorical variables, we selected three types of tasks such as retrieving value as a low-level task; comparing and ranking as a high-level task. For the comparing and ranking task, two different types of questions were asked: the tasks to consider absolute values such as frequency and tasks that consider relative values such as percentage. Therefore, for one visualization 5 different questions were generated.

[1] `https://www.mturk.com`

[2] `https://purdue.qualtrics.com/SE/?SID=SV_9YX7LCgsiwv0Voh`

For gatherplots, our interest is more about the difference between questions considering absolute values and relative values. The five types of questions are as follows:

- **Type 1:** retrieve value considering one subgroup
- **Type 2:** comparing of absolute size of subgroup between groups
- **Type 3:** ranking of absolute size of subgroup between groups
- **Type 4:** comparing relative size of subgroup between groups
- **Type 5:** ranking relative size of subgroup between groups

To reduce the chance of one chart being optimal by luck for specific task, two charts of same problem structure were provided. Eventually, the resulting questions were 10 for each participant. Each question was followed by the question asking confidence of estimation with a 7-point Likert scale, and the time spent for each question was measured.

Hypotheses

We believe that different types of tasks will favor from different type of layouts. Therefore our hypotheses are as follows:

H1 For retrieving value considering one subgroup (Type 1), absolute, relative, both mode reduces the occurrence of the error than jitter mode.

H2 For tasks considering absolute values (Type 2 and 3), the absolute mode reduces the error.

H3 For tasks considering relative values (Type 4 and 5), the relative mode reduces the error.

Results

The results were analyzed with respect to the accuracy (correct or incorrect), time spent, and confidence of estimation. Based on our hypotheses, we analyzed the different modes of layout for each type of question: retrieve value, absolute value task, and relative value task.

Accuracy

The number and percentage of participants who answered correct and incorrect answers are shown in Figure 12. Eventually, we had 42 participants for jitter, 56 participants for absolute, 56 participants for relative, and 57 participants fro both mode.

As the measure for each question was either correct or incorrect, a logistic regression was employed using PROC LOGISTICS in SAS. For the retrieving-value task (Type 1), both the the absolute mode and relative mode had significant main effects (Wald Chi-Square = 18.58, $p < 0.01$, Wald Chi-Square = 21.05, $p < 0.01$, respectively) with a significant interaction effect (Wald Chi-Square = 19.53, $p = 0.03$) (H1 confirmed). For absolute-value tasks (Type 2 and 3), both the the absolute mode and relative mode had significant main effects (Wald Chi-Square = 10.35, $p < 0.01$, Wald Chi-Square = 10.35, $p < 0.01$, respectively) with a significant interaction effect (Wald Chi-Square = 4.31, $p = 0.03$) (H2 confirmed). For relative-value tasks (Type 4 and 5), only the relative mode had a significant effect (Wald Chi-Square= 5.10, $p = 0.02$) (H3 confirmed).

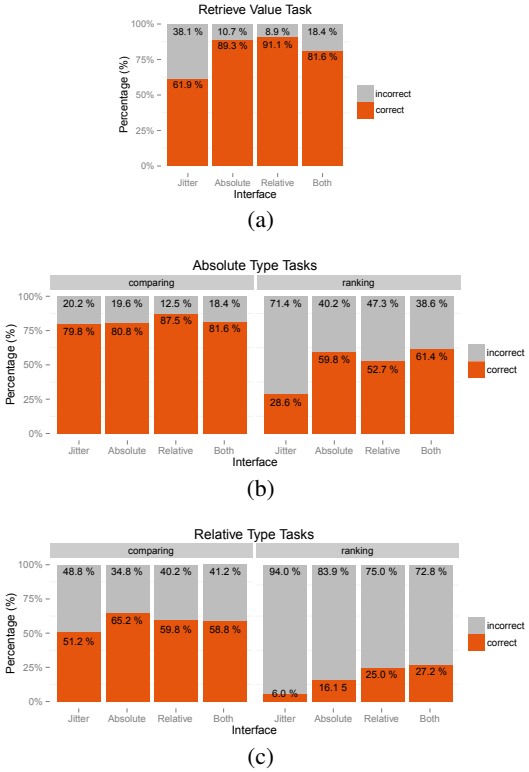


Figure 12. (a) The percentage of participants who have got the answer correct for retrieving value task. (b) The percentage of participants who have got the answer correct for absolute type tasks for comparing and ranking. (c) The percentage of participants who have got the answer correct for relative type tasks for comparing and ranking.

Time spent

The time spent (in seconds) for each question was compared using mixed-model ANOVA with repeated measures. For the retrieving-value task, on average, the time spent (sec) for each interface was for jitter (44.26), absolute (56.84), relative (52.45), and both (56.57). There was no significant difference between interfaces ($p > 0.05$ for all cases).

For the absolute-value task (Type 2 and 3), on average, the time spent (sec) for each interface was for jitter (30.74), absolute (32.3), relative (33.6), and both (47.91). The interface had a significant main effect ($F(3, 207) = 11.5, p < 0.01$). However, when we conducted pairwise comparisons with adjusted p-values using simulation, the only significant difference in time spent was when using the both interface which took longer ($p < 0.01$ for all comparisons).

For relative-value task (Type 4 and 5), on average, the time spent for each interface was for jitter (26.6), absolute (31.12), relative (31.38), and both (46.78). The interface had a significant main effect ($F(3, 207) = 10.12, p < 0.01$). However, when we conducted pairwise comparisons with adjusted p-values using simulation, the only significant difference in time spent was when using the both interface which took longer ($p < 0.01$ for all comparisons).

Confidence

The 7-point Likert-scale rating was used for the level of confidence on their estimation. For the value-retrieving task (Type 1), Kruskal-Wallis non-parametric test revealed that the type of interface had significant impact on the confidence level ($\chi^2(3) = 74.57 p < 0.01$). The mean rating for each interface was for jitter (4.8), absolute (6.3), relative (6.0), and both (6.25). A post-hoc Pairwise Wilcoxon Rank Sum test was employed with Bonferroni correction to adjust errors. The jitter interface was significantly lower than the other three modes ($p < 0.01$ for all cases). There was no difference between absolute, relative, and both interfaces.

For absolute-value tasks (Type 2 and 3), Kruskal-Wallis non-parametric test revealed that the type of interface had significant impact on the confidence level ($\chi^2(3) = 18.32, p < 0.01$). The mean rating for each interface was jitter (5.4), absolute (5.7), relative (5.0), and both (5.8). A post-hoc Pairwise Wilcoxon Rank Sum test was employed with Bonferroni correction to adjust errors. The interface with both mode was significantly higher than relative and jitter mode ($p < 0.01$ for both), however no difference with the absolute mode. The interface with absolute mode was significantly higher than relative and jitter mode ($p < 0.01$).

For relative-value tasks (Type 4 and 5), Kruskal-Wallis non-parametric test revealed that the type of interface did not have significant impact on the relative tasks ($\chi^2(3) = 4.1, p = 0.2$). The mean rating was jitter (4.7), absolute (4.9), relative (4.9), and both (4.8).

One possibilities for result is that relative task might be harder than others. The low correct percentage of questions are also shown in Figure 12(c). To see that, we have tested the confidence level among task types. Kruskal-Wallis non-parametric test revealed that the type of task had significant impact on the confidence level ($\chi^2(2) = 148.1, p < 0.01$). The mean rating for retrieving value (5.9), absolute (5.5), and relative (4.8). The post-hoc Pairwise Wilcoxon Rank Sum test was employed with Bonferroni correction to adjust errors showed that all three task types have significantly different ($p < 0.01$ for all cases).

DISCUSSIONS

Scalability

As the dataset tend to become large size, the scalability of visualization becomes an important issue. There are two main tasks related to the scatterplots, which are an overview of correlation and detection of outliers. Gatherplots are effective in showing the overview as the dataset becomes large. However as the dataset increases, the dot size shrinks. And the detection of outliers becomes less plausible. In this sense, gatherplots are not scalable to the large datasets. It is more applicable to the dataset of small to medium size, with multi-dimensionality. Also as the dataset becomes large, individual object identification becomes less relevant, and the gatherplots resembles histograms itself, which is similar with Jean-Francois et al. [20]. One particular worst case for gatherplots are when there are severe concentration of value over specific values. As the dot size and visual space for clusters are same over the categorical values, this makes overall dots size very small. Relative mode by user interaction can alleviate this problem, because it allows small outliers to become large enough to be visible.

Evaluation Challenges

Conducting quantitative experiments with visualizations is challenging. First, as visualizations support various cognitive tasks, designing a representative task is challenging. Scatterplots supports diverse types of tasks such as detecting correlation, clusters, or outliers. In this experiment we decided to test a particular case with categorical data, which has distinctive views compared to conventional scatterplots. Due to the categorical data structure, the task types was also limited to a specific context. Although it is a narrow case, the purpose of this study was to show the effectiveness of different layout modes in a quantitative way. The results indicated that the users could understand the visualization and accomplish the task that should be supported. However, we could also observe that the difficulty level was different for each task type. In general, ranking tasks were more difficult than comparing tasks and questions asking about relative values were more difficult than those of absolute values. Therefore, maintaining similar difficulty level among tasks should also be considered while designing tasks. Second, when designing a study to evaluate a new technique, it is challenging to design a proper baseline condition. As visualizations have several features, the baseline should be selected to be different only for key features. If various parts are different, it is hard to understand what part has affected user performance. In our study we selected scatterplots with jittering as baseline as it is 1) a technique for the scatterplots for overplotting, 2) it maintains individual objects, and 3) well-known technique. So it provides base-line for the performance. But it would be also desirable to compare the performance with a purpose-specific technique, such as histograms or hieraxes.

CONCLUSION AND FUTURE WORK

We have proposed the gatherplots, an extention of scatterplots, which enable overview without clutter for multidimensional data including categorical variables. While gatherplots are optimal for categorical variables, it can also be used to ameliorate overplotting caused by continuous ordinal variables. We discussed several aspects of gatherplots including layout, coloring, tick format, and matrix formations. We also evaluated the technique with a crowdsourced user study showing that gatherplots are more effective than the jittering, and absolute and relative mode serves specific types of tasks better. Finally, in-depth feedback from an expert review involving visualization reviewers revealed several limitations for the gatherplots technique. We addressed these weaknesses and suggested possible remedies.

We believe that gathering is a general framework to formulate the transition of overlapping visualization to space-filling visualization without sense of individual objects. In the future, we plan on studying the application of this framework to other visual representations to explore novel visualizations.

For example, parallel sets can be reconstructed to render individual lines instead of block lines, which would enable combining both categorical and continuous variables. Gathering also enables mixing nominal variables and ordinal variables in a single axis. This can be pursued further, for example in a gathering lens that gathers underlying objects according to a data property. If we apply this lens to selected boundary in crowded region of scatterplots, the underlying distribution of that region can be revealed.

ACKNOWLEDGMENTS

[Anonymized for double-blind review.]

REFERENCES

1. B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics*, 21(4):833–854, 2002.
2. A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J.-D. Fekete. GraphDice: A system for exploring multivariate social networks. *Computer Graphics Forum*, 29(3):863–872, 2010.
3. D. Carr, W. Nicholson, R. Littlefield, and D. Hall. Interactive color display methods for multivariate data. In *Naval Research Sponsored Workshop on Statistical Image Processing and Graphics*, pages 215–250, 1983.
4. H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.
5. W. S. Cleveland and M. E. McGill. *Dynamic graphics for statistics*. CRC Press, 1988.
6. W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 30 1985.
7. L. Cosmides and J. Tooby. Are humans good intuitive statisticians after all? Rethinking some conclusions from the literature on judgment under uncertainty. *Cognition*, 58(1):1–73, 1996.
8. 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.
9. A. Dix and G. Ellis. By chance - enhancing interaction with large data sets through statistical sampling. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*, pages 167–176, 2002.
10. G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, Nov. 2007.
11. N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008.
12. N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, 2010.
13. J.-D. Fekete and C. Plaisant. Interactive information visualization of a million items. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 117–124, 2002.
14. Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the IEEE Conference on Visualization*, pages 43–50, 1999.
15. S. Haroz and D. Whitney. How capacity limits of attention influence information visualization effectiveness. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2402–2410, 2012.
16. J. A. Hartigan and B. Kleiner. Mosaics for contingency tables. In *Computer Science and Statistics: Proceedings of the Symposium on the Interface*, pages 268–273. Springer, 1981.
17. S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: Visualizing theme changes over time. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 115–123, 2000.
18. H. Hofmann, A. P. J. M. Siebes, and A. F. X. Wilhelm. Visualizing association rules with interactive mosaic plots. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pages 227–235, 2000.
19. S. Huron, R. Vuillemot, and J.-D. Fekete. Visual sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, 2013.
20. J.-F. Im, M. J. McGuffin, and R. Leung. Gplom: The generalized plot matrix for visualizing multidimensional multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2606–2614, 2013.
21. R. Kosara, F. Bendix, and H. Hauser. Parallel sets: interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, 2006.
22. A. Mayorga and M. Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 19(9), Sept. 2013.
23. B. McDonnel and N. Elmqvist. Towards utilizing GPUs in information visualization: A model and implementation of image-space operations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1105–1112, 2009.
24. L. Micallef, P. Dragicevic, and J.-D. Fekete. Assessing the effect of visualizations on Bayesian reasoning through crowdsourcing. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2536–2545, 2012.

25. G. Paolacci, J. Chandler, and P. Ipeirotis. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5):411–419, 2010.

26. B. Shneiderman, D. Feldman, A. Rose, and X. F. Grau. Visualizing digital library search results with categorical and hierarchical axes. In *Proceedings of the ACM Conference on Digital Libraries*, pages 57–66, 2000.

27. B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

28. S. S. Stevens. On the theory of scales of measurement. *Science*, 103(2684):677–680, June 1946.

29. M. Trutschl, G. Grinstein, and U. Cvek. Intelligently resolving point occlusion. In *Proceedings of IEEE Symposium on Information Visualization*, pages 131–136, 2003.

30. E. R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, 1983.

31. J. M. Utts. *Seeing Through Statistics*. Duxbury Press, 1996.

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

33. W. Willett, S. Ginosar, A. Steinitz, B. Hartmann, and M. Agrawala. Identifying redundancy and exposing provenance in crowdsourced data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2198–2206, 2013.

34. S. Zhai, W. Buxton, and P. Milgram. The partial-occlusion effect: utilizing semitransparency in 3D human-computer interaction. *ACM Transactions on Computer-Human Interaction*, 3(3):254–284, 1996.