Binning Strategies and Related Loss for Large Data

Susan VanderPlas[1], Karsten Maurer[1], Heike Hofmann[1,2]
[1]Department of Statistics, [2]Human Computer Interaction
Iowa State University

March 30, 2013

Abstract

Dealing with the data deluge of the Big Data Age is both exciting and challenging. The demands of large data require us to re-think strategies of visualizing data. Plots employing binning methods have been suggested in the past as viable alternative to standard plots based on raw data, as the resulting area plots tend to not be affected by increases in data as much. This comes with the price of loss of information inherent to any binning scheme. In this paper we discuss properties of two commonly used binning algorithms. We define loss of information in the specific setting of two dimensional displays, provide readily applicable tools for loss evaluation and discuss the two binning schemes with respect to their loss in the framework of a simulation and two case studies.

1 Introduction

Technological advances have facilitated collection and dissemination of large data more and more as records are digitized and our lives are increasingly lived online. According to estimates published by the IDC in 2009 the worldwide digital content doubles about every 18 months – in 2011 the digital amount was estimated to be at 1.8 zeta bytes (1 ZB = 2^{70} bytes $\approx 10^{21}$ bytes). This "Data Deluge" of the Big Data Age (NY Times, Feb 2012) poses exciting challenges to data scientists everywhere:

> "It's a revolution ... The march of quantification, made possible by enormous new sources of data, will sweep through academia, business and government. There is no area that is going to be untouched"

(Gary King, Harvard Institute).

Data sets with millions of records and thousands of variables are not uncommon. Such data sets are often too large to fit into a single computer's working memory, and single-user machines cannot easily deal with them. We can employ database tools to extract relevant variables, thereby vertically subsetting the data. However, with millions of records, it can be very difficult to work with data even at that level.

Friedman (1997) proposed in his paper on data mining and statistics that "Every time the amount of data increases by a factor of ten, we should totally rethink how we analyze it". The same holds for visualizations. With a 100-1000 fold increase in the amount of data, the utility of some of our most used graphical tools, such as scatterplots, deteriorates quickly (Unwin et al., 2006).

Area plots, such as histograms, do not tend to be as affected by increases in the amount of data as plots that utilize raw data. By using binning strategies and the principles for displaying information in area plots, scatterplots can again become useful instruments for large data settings (Unwin et al., 2006).

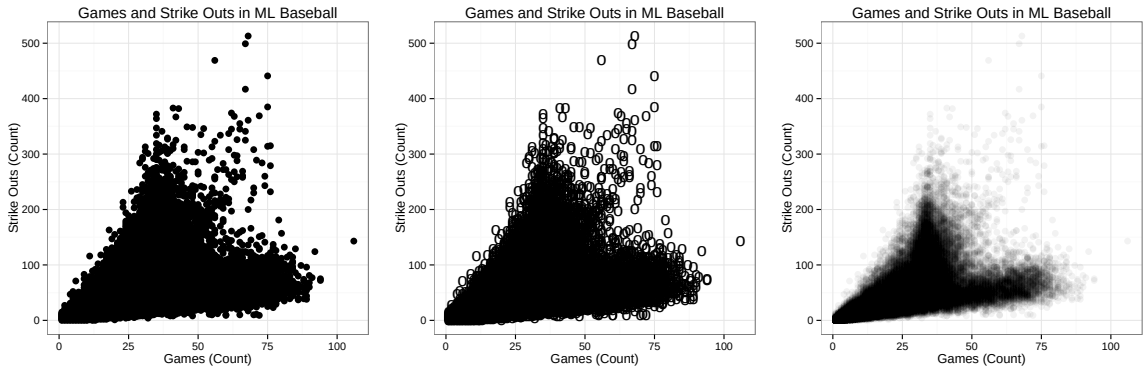
In this paper we describe first the problem scatterplots are exposed to in large-data situations. We discuss two different binning methods and introduce the *loss of information* inherent to binning. We conclude with a detailed example.

2 Scatterplots for Large Data Sets

In the case of medium sized data scatterplots are great tools for showing relationships in two dimensions. For large data, scatterplots suffer from over-plotting – i.e. more and more points are drawn in close-by places, thereby masking relevant structure. Figure 1 shows an example taken from baseball statistics. The scatterplot shows 139 seasons (from the years of 1871 – 2009) of pitching statistics for every baseball pitcher as published in Sean Lahman's Baseball database (`http://www.seanlahman.com/baseball-archive/`). The number of games played in a season is plotted versus number of strikeouts a pitchr threw over the course of a season. While the data set is only medium sized with 39745 observations, it already shows some of the break-down patterns scatterplots experience with large data. Figure 1 shows a traditional scatterplot on the left. Each observation is drawn with a filled circle. A triangular structure is apparent with some outliers at a medium number of games and high number of strikeouts. Tukey (Tukey, 1977) suggested the use of open circles (see Figure 1b) to mitigate the problem of over-plotting. Open circles make points visible that are close together. This technique is not suitable to determine the magnitude, but gives more information than is available with filled points. A modern alternative to open circles is alpha-blending (see Figure 1c). Alpha blending provides more frequency information, allowing a distinction of higher-resolution density.

All of these methods fall short in the example. As can be seen in Figure 1, strategy (a) is the least effective, as it provides information about the outliers and range of the data but cannot provide any point density information. Tukey's open circles (b) help some, but are as prone to over-plotting as solid circles when the data set is large. Alpha blending (c) highlights the structure, but minimizes the visual impact of outliers. The data set is large enough that neither alpha-blending nor open circles are completely effective, and so we must pursue a different strategy which can provide better information about the relative density of points at a given location.

One approach to reduce the graphical complexity is to bin the displayed data. This has the additional advantage of reducing the size of the stored data, as only the bin centers and



(a) Overplotted data

(b) Tukey-style open circles

(c) Alpha blending

(d) Hexagonal binning

(e) Scatterplot of Binned Data

Figure 1: Scatterplots of Games versus Strikeouts in Major League Baseball, using different strategies of dealing with the issue of over-plotting: (a) uses standard, opaque, filled circles, (b) uses Tukey's recommended open circles, and (c) uses filled circles with alpha-blending (α=0.05). Plots (d) and (e) show hexagonal binning strategies with frequency mapped to color and area respectively

the frequency of points which correspond to that bin must be stored.

Histograms are an example of plots which use binned variables. To extend histograms to a graph which can represent the joint distribution between two variables, it is a natural step to form a tesselated grid on a two dimensional Cartesian plane and use some other attribute (color, 3D rendering) to provide joint density information within each grid cell, known as a tile. A *binned scatterplot* uses shading to provide frequency information, with tiles (rather than bars in a histogram) centered at the bin center, rather like a two-dimensional histogram viewed from above.

Methods commonly used to display binned variables include sunflower plots (Cleveland and McGill, 1984), modified scatterplots, and kernel density smoothing of tonal variation (Theus, 2006). Sunflower plots are scatterplots of binned data, where the symbol used for the bin increases in complexity in proportion to the number of points in that bin. Sunflower plots are particularly useful when the number of points in each bin remains reasonably small.

When the point area is scaled in proportion to frequency, scatterplots can be used to display the binned data as well. When points are filled circles, these plots are also known as "bubble plots", which were first used by William Playfair (Playfair, 1786; Playfair et al., 2005). Kernel density smoothing can be used to vary α or color according to a smoothed density, providing features similar to binned scatterplots or α-blended scatterplots in a more smooth, continuous fashion. These estimates require parameter tuning and may hide gaps in the data by over-smoothing while simultaneously de-emphasizing outlying points.

As alternatives, Figure 1 (d-e) contains examples of a hexagonally binned scatterplot with frequency encoded as color (d) and a "bubble plot" (referred to as a modified scatterplot) with frequency encoded as point size (e). The hexagonal tiles and the modified scatterplot are more effective at displaying the shape of the joint density and preserving outliers than any of the scatterplots shown in Figure 1 (a-c). The binned scatterplot is less prone to the Hermann-grid illusion than the bubble plot, particularly for binned data, where bin centers usually fall on a grid.

Only in the binned scatterplot and the bubble plot the inner structure of the data becomes apparent: the joint density consists two distinct ridges following two lines with very different slopes. The lower slope corresponds to the modern average strike out rate of pitchers of just under one strike-out per game. The other line has a slope of about four times that rate. This high rate is also associated with fewer games played. Closer investigation of other, related variables reveals that this high strike-out rate corresponds mainly to historic pitchers with much shorter seasons (in 1876 only 70 games were played in a season, as opposed to 162 in 2009), and qualitatively different balls and bats.

For extremely large data sets, binned scatterplots are a more useful visualization of two-dimensional density information than the scatterplot, and are less computationally demanding, as not every single point in the data set has to be rendered separately.

As with histograms, the width of bins (or the number of bins) is an important factor in the detail of the binned data and the resulting plot: if the bin width is too small in comparison to the amount of data available, there is little advantage to binning, but if the bin width is too large, interesting features of the joint distribution may be obscured by over-smoothing.

3 Binning Data

We will only consider binning in two dimensions, X and Y. The algorithms we discuss are immediately applicable to higher dimension, but we do not feel that the paper would benefit from a more general discussion. Binning in dimensions X and Y provides us with a more condensed form of the data that ideally preserves both the joint distribution as well as the margins, while reducing the amount of information to a fraction of the original.

Binning is a two-step procedure: we first assign each data tuple (x, y) to a bin center (x^*, y^*), and in a second step we find the frequency of each bin center, resulting in triples of the form (x^*, y^*, c), where c is the number of all observations assigned to bin center (x^*, y^*).

We will proceed with rectangular bins for simplicity, but other binning schemes, such as those which utilize hexagonal bins are also common (Carr et al., 1987). Rectangular bins are advantageous because bins in x and y are orthogonal to each other, thus, we can present the one-dimensional case which will easily generalize to two dimensions.

3.1 Standard Rectangular Binning Algorithm

Let $x_{(i)}$ correspond to the ordered values of X, where $i = 1, ..., n$ and n is the number of total observations. Then $x_{(1)}$ and $x_{(n)}$ then denote the observed minimum and the maximum of X, respectively.

We then define a set of bin centers $\{x_j^*\}$, with $1 \leq j \leq n_X$, where n_X is the overall number of bins in dimension X. Binning can either be determined by the number of bins, n_X, or, equivalently, by the bin width, b_X. For standard binning the relationship between these two parameters is given as

$$b_X = (x_{(n)} - x_{(1)})/n_X$$

Mathematically, binning corresponds to a function $b(.)$ that assigns to a value x the closest bin center $x^* \in \{x_j^*\}, j \in \{1, ..., n_X\}$, i.e. the binning function $b(.) : x \to x^*$ is defined as

$$x^* = b(x) = b_X \cdot \lfloor (x - O_X)/b_X + b_X/2 \rfloor + O_X$$

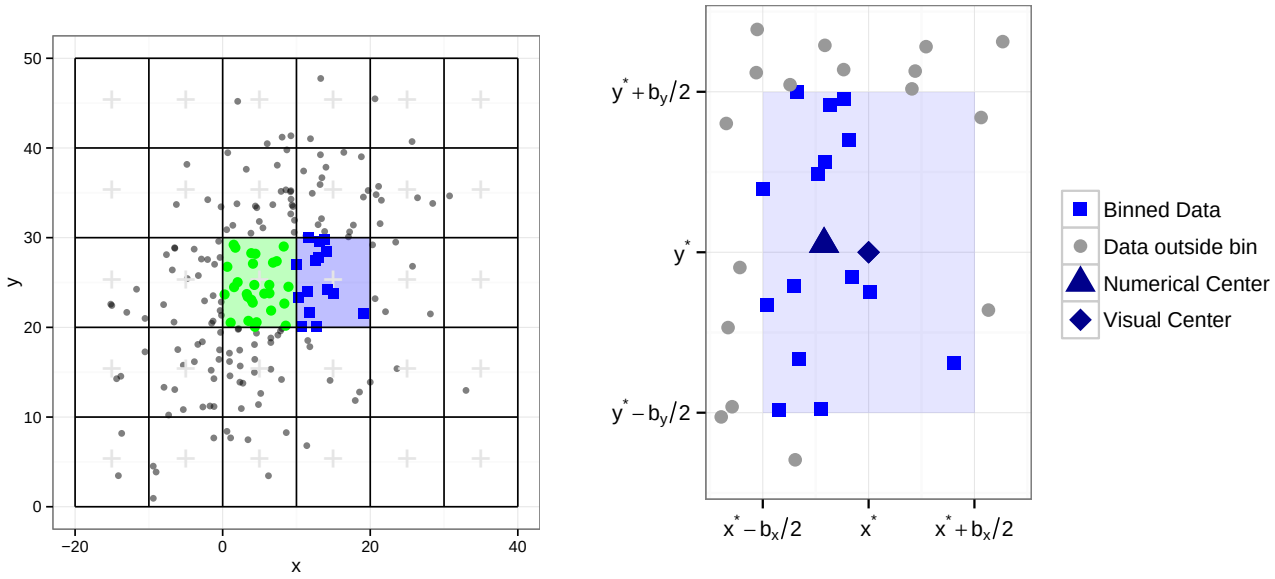
where O_X is the center of the first bin and $\lfloor . \rfloor$ denotes the floor function, $\lfloor x \rfloor$ is the largest integer value smaller than x.

For illustration, let us assume that we have observed values 1.23, 1.55, and 2.35, and bin centers $\{1, 2, 3\}$, corresponding to a bin width of $b_X = 1$ and a first bin center $O_X = 1$. Intuitively it is clear, that value 1.23 is assigned to bin center 1, while values 1.55 and 2.35 are assigned to bin center 2. Applying equation (1) to value 1.23 we get: $1 \cdot \lfloor (1.23-1)/1+0.5 \rfloor +1$ $= \lfloor 0.73 \rfloor + 1 = 0 + 1 = 1$. Similarly, 1.55 is assigned to $\lfloor 0.55 + 0.5 \rfloor + 1 = 1+1 = 2$.

Alternate binning parameterizations include dividing up the range of data by a specified number of bins of equal width, but even these strategies fit into the framework of assigning values to the closest bin center.

Figure 2 provides an illustration of the binning process.

The method described above is the standard binning algorithm. As a different approach, we define the random binning algorithm below.



(a) Standard Algorithm

(b) Bin Attribute Diagram

Figure 2: Points are assigned to bins centered at (5, 25) an (15,25). Points in the standard algorithm are all contained within the bin region, where points in the random algorithm may be outside the direct region.

3.2 Random Binning Algorithm

An interesting variation which results in statistically nice properties is a random, non-deterministic bin function $b^r(\cdot)$ which maps value x to one of several bin centers x^* with probability p. In this paper, we will consider the simplest case of just two bins, so that without loss of generality we can assume that x lies between bin centers x_j^* and x_{j+1}^*. The bin function assigns x to bin center x^* with a probability depending on the distance to that bin center; the closer a value is to a bin center, the higher the probability that the value is assigned to this bin center. More formally,

$$b^r(x) = \begin{cases} x_j^* & \text{with probability } p = (x_{j+1}^* - x)/b_X \\ x_{j+1}^* & \text{with probability } 1 - p = (x - x_j^*)/b_X \end{cases} \quad (1)$$

This method is easily extensible to also map x into one of more than two bins and can accommodate non-uniform bin sizes, unlike the standard algorithm.

Figure 2 illustrates one possible outcome of the random binning process.

The deterministic standard binning algorithm is an example of a "direct" binning algorithm, in which all points are assigned with weight one to the nearest bin. "Linear" binning

(Theus, 2006) is a *computationally intensive* alternative to direct binning in which adjacent bins are assigned a weight depending on the distance from the point to that bin, where all weights sum to one. With large data sets, the calculations required for linear binning become unwieldy, but the random binning algorithm can be considered an approximation to linear binning. Specifically, the expectation of the random binning algorithm is the same as for linear binning. In the next section, we examine the loss of visual and numerical information due to binning.

4 Loss due to Binning

While binning data has some notable advantages, it does result in loss of some information. Using a small number of bins removes much of the relevant information in the data, while an extremely large number of bins may not reduce the data size or complexity to a sufficient degree. Figure 3 gives an overview of a data set and binned representations using different numbers of bins, demonstrating the loss of information with increasing bin size.

In the first set of binned data, the bins and the points on the scatterplot are nearly identical, but the scatterplot contains information about overlapping points. The second and third sets of binned data, with bin width=0.25 and 0.50 respectively, show higher-level summaries of the data that contain some numerical and visual loss but which may also provide more visually accessible information about the shape of the two-dimensional density between x and y. The fourth set of binned data with bin width = 1.0 is nearly unrecognizable, because the bins are large enough that they provide very little additional information.

Loss of information occurs on multiple levels during the binning and rendering process. We distinguish three sources:

- *Visual Loss* is an implicit loss due to the rendering; visually we perceive the center of a bin to be the center of the tile. This visual center will generally different from the numerical center, or the mean of the data corresponding to that bin, which can be thought of as the center of mass. Thus visual loss is the difference between the visual center (x^*, y^*) and the numerical center $(\bar{x}, \bar{y})$. This causes some bias in the perception.

- *Numerical Loss* occurs when points in the data set are reduced to a single point at the mean, i.e. this is the collective difference between all data points (x_i, y_i) and their corresponding numerical centers $(\bar{x}, \bar{y})$.

- *Frequency Loss* results from our inability to render the frequency information accurately. Different rendering methods for the frequency information will lead to different losses. For the remainder of the paper we will assume that we are using colors in binned scatterplots to represent this information. The number of colors that can be differentiated and mapped back to frequency information accurately is fairly limited. Note that even though this loss will turn out to be substantial, it is, in fact, a huge gain with respect to the original scatterplot, where frequency information is masked in large data situations due to over-plotting of points.

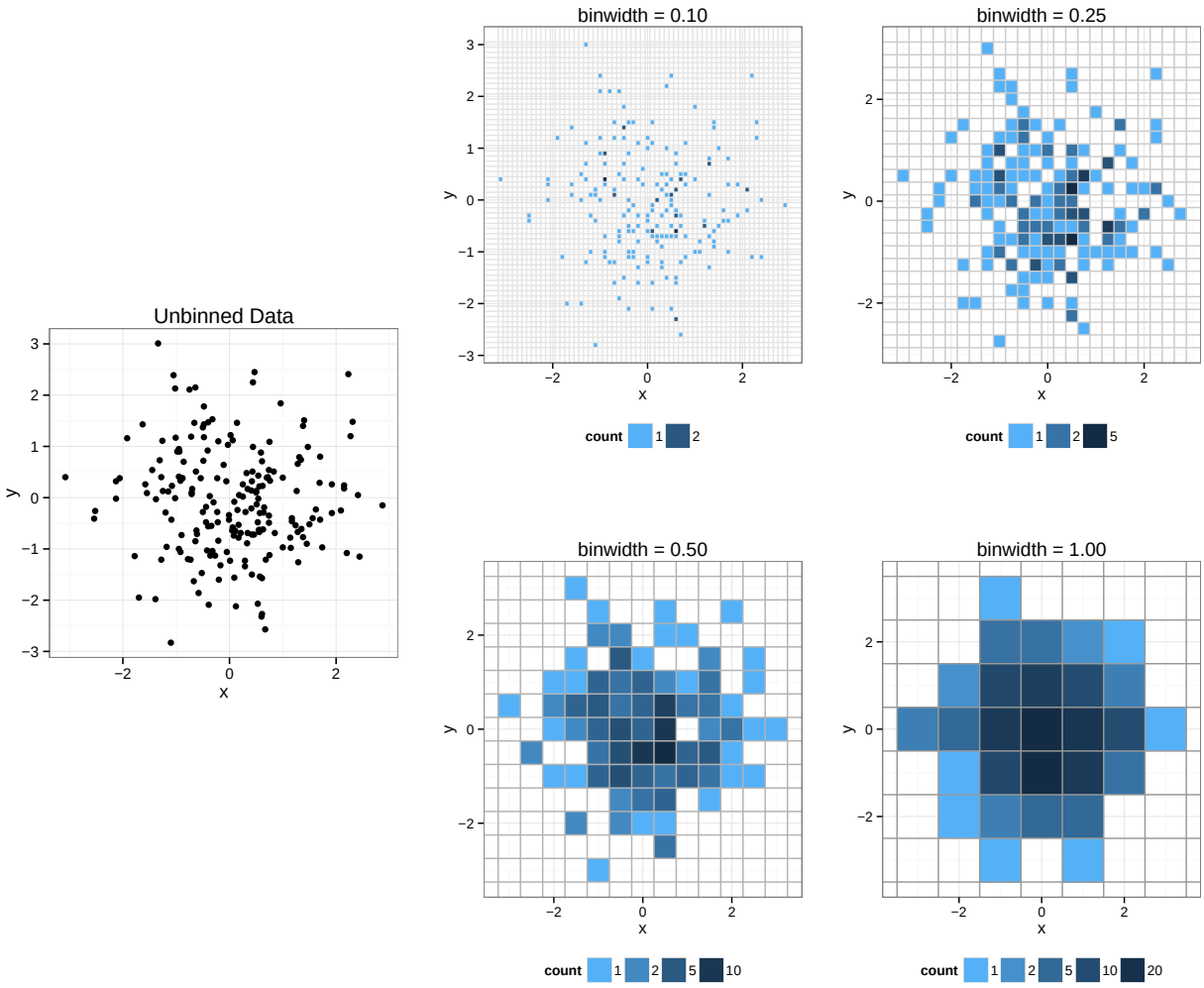


Figure 3: Series of scatterplots showing the original data (scatterplot, left), and versions of the binned data for different bin widths. The visual loss from binning at 0.1 is minimal, while a bin width of 1 gives a rough approximation.

We explore the different types of loss in the remainder of this section in more detail. For clarity we will again be specifying loss using notation for a one dimensional binning process. This notation can then be extended to higher dimensions in applications.

4.1 Numerical and Visual Loss

Reducing data points $\mathbf{x}$ to the visual bin centers x^* results in a loss of information, as there is a reduction in the variability of the data when data points are transformed from individual values to bin centers on a grid. Let us denote the difference between point x_i from its bin center as $S_i = |x_i - b(x_i)|$ for each point in the data set $i = 1, ..., n$. The total loss from binning an entire data set is defined as

$$L_s(\mathbf{x}) = 1/S_\emptyset^2 \cdot \sum_{i=1}^{n} S_i^2$$

where $S_\emptyset^2$ is the loss which results when the entire data set is reduced to a single bin, $S_\emptyset^2 = \sum_i (x_i - \tilde{x})^2$ for the single bin center $\tilde{x}$. If the visual bin centers are the same as the numerical centers for each bin, then any loss could be strictly attributed to difference between points and the bin centers. The problem is that the visual center and the numerical center for bins will rarely match, which introduces visual bias to the graphical representation of the data, as we perceive the bin center to be the physical middle of the symbol representing the data. Therefore we distinguish two contributions to the total loss. Numerical loss results from the binning process as points are reduced to the numerical center point and is thus the sum of numerical losses for individual points ($\sum_{i=1}^{n} L_i^{\mathrm{N}}$). Visual loss is due to the difference between the numerical centers and the visual centers and is thus the sum of visual losses for each bin ($\sum_{j=1}^{N_x} L_j^{\mathrm{V}}$). This discrepancy is particularly clear as demonstrated in Figure 4, in which the visual center of the bin varies significantly from the numerical center of the data contained within the bin under the standard algorithm.

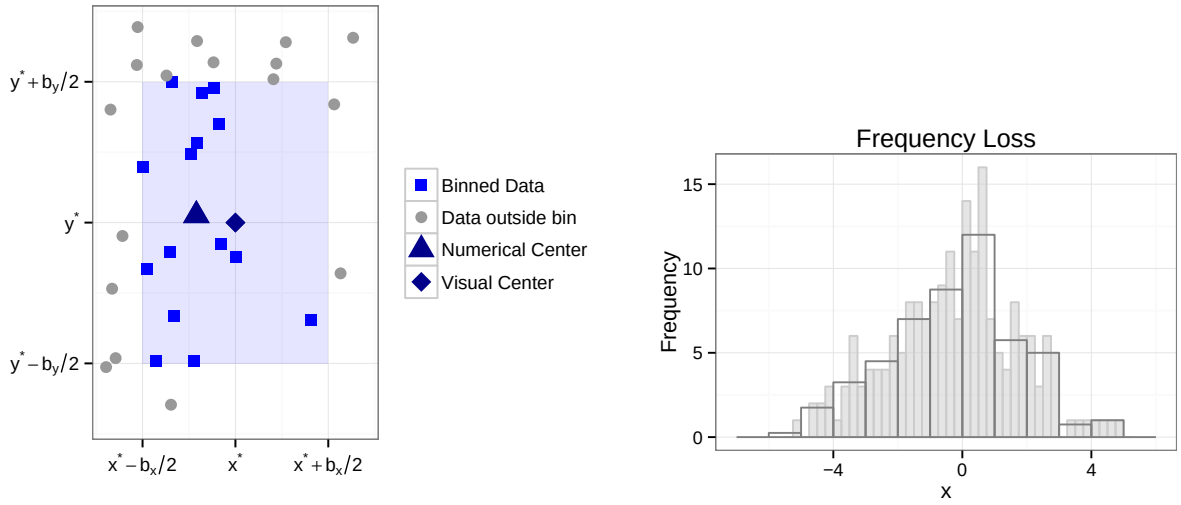
Fortunately we can show that the total loss perceived is partitioned by these two components of loss, and it holds (see appendix A.1 for a proof):

$$L_s(\mathbf{x}) = 1/S_\emptyset^2 \cdot \left(\sum_{i}^{n} L_i^N + \sum_{j}^{N_x} L_j^V \right) \tag{2}$$

Note that the discrepancy between the visual center of the bin and the numerical center of the data within the bin occurs in both binning algorithms. The decomposition also holds for the random binning algorithm as well as the standard binning algorithm.

It is worth noting that the loss function is a monotonically increasing function of binwidth. An interesting feature of the random binning algorithm is that the increase in loss that we experience by doubling the bin width is deterministic, even though the assignment of bins is not. This holds for repeated doubling of the bin width. A proof of this can be found in appendix A.2. Doubling bin widths is, particularly in the framework of visualizations, a very natural step, making this an important property of the random binning.

The loss as a function of bin width was computed for both algorithms for the baseball data and is shown in Figure 6.



(a) Spatial Loss

(b) Frequency Loss

Figure 4: Types of loss due to binning: (a) The visual center of the bin is different from the numerical center of the bin, leading to two different sources of loss. (b) Frequency loss associated with a change of bin width to 1 from 0.25 in the example of Figure 3.

4.2 Loss of Frequency Information

Loss of frequency information in a scatterplot happens because of our inability to render elements of a third dimension efficiently. By employing a color scheme for filling tiles according to the observed frequencies corresponds to a binning of the frequencies into a few different levels, with losses stemming from the same source as the previously discussed numerical loss. We will assume that we have a set of size n_C of frequency levels $\{c_k^*\}$ with $1 \leq k \leq n_C$. An observed frequency c_j with $1 \leq j \leq N_x \times N_y$, where N_x and N_y are the number of bin divisions in the X and Y directions respectively, is assigned to the closest frequency level $c_k^* = b(c_j)$. The *unscaled* frequency loss is then defined as

$$L_{\text{Freq}} = \sum_{j=1}^{N_x N_y} (c_j - b(c_j))^2 \tag{3}$$

The loss of frequency information due to binning results from tiling over larger areas, thus results from *smoothing* the frequency variation (see Figure 4).

Frequency data consists of counts, which most commonly exhibit skew densities, i.e. there are usually a lot of cells with small cell counts and a few cells with extremely large counts.

A log transformation therefore produces a more symmetric distribution of frequency information, increasing perceptual resolution. This is consistent with the Weber-Fechner law which suggests that increased stimulus intensity is perceptually mapped on the log scale (Goldstein, 2007). Using a logarithmic mapping of frequency to the color or size aesthetic provides a more natural perceptual experience and simultaneously increases the perceptual resolution of the graph. This suggests that frequency loss be calculated as

$$L_{\log \text{Freq}} = \sum_{j=1}^{N_x N_y} \left(\log(c_j + 1) - \log(b(c_j) + 1)\right)^2 \tag{4}$$

Note that we have added one count to every cell before taking the logarithm to avoid problems with empty cells.

This numerical assessment of frequency loss does not account for limitations in human perceptual ability. Research suggests that under optimal conditions, we can effectively compare about seven colors (Healey and Enns, 1999), which provides a physical upper limit on the amount of frequency variation we can perceive. As a result, a binning width which produces fewer than seven frequency categories is preferable, while minimizing numerical loss within that constraint.

To examine this further, we simulated points from three different distributions: Uniform, Normal, and Exponential to determine under what circumstances log frequency scaling is performing better than linear frequency scaling. Figure 5 gives an overview of this simulation. In the top row binned scatterplots with a color scheme based on linear frequency scaling is used, the bottom row shows the same data based on log frequency scaled colors.

For uniformly distributed data (right column) the choice of color scheme does not have a huge impact, but under both the normal and exponential distribution, log frequency scaling results in categories which account for more uniform proportions of the data (as can be seen by the color strips in the second and third row of Figure 5). This is desirable because it provides more discernible frequency information.

Frequency loss is much more difficult to normalize, as we must balance the partial loss of frequency information due to over-plotting with the total loss of frequency information that results if we use only a single bin to categorize the data. For the purposes of numerical assessment of frequency loss, we calculate the baseline frequency loss as the loss corresponding to a minimal bin width given by the smallest non-zero absolute difference of successive values in the original data set, as shown in Figure 3.

Traditional Scatterplots and Frequency Information: Using a minimal bin width, a binned scatterplot is comparable to a standard scatter plot, with bins shaded in a binary manner, as each unique observed value is located in a different bin. Alpha blending as used in Figure 1c extends the binary shading of a standard scatterplot to an implicit shading according to frequency. The shading is implicit because the range of frequency information is not scaled to the range of shading values, so that maximum color saturation is reached well before the maximum frequency, truncating the perceivable frequency information. By

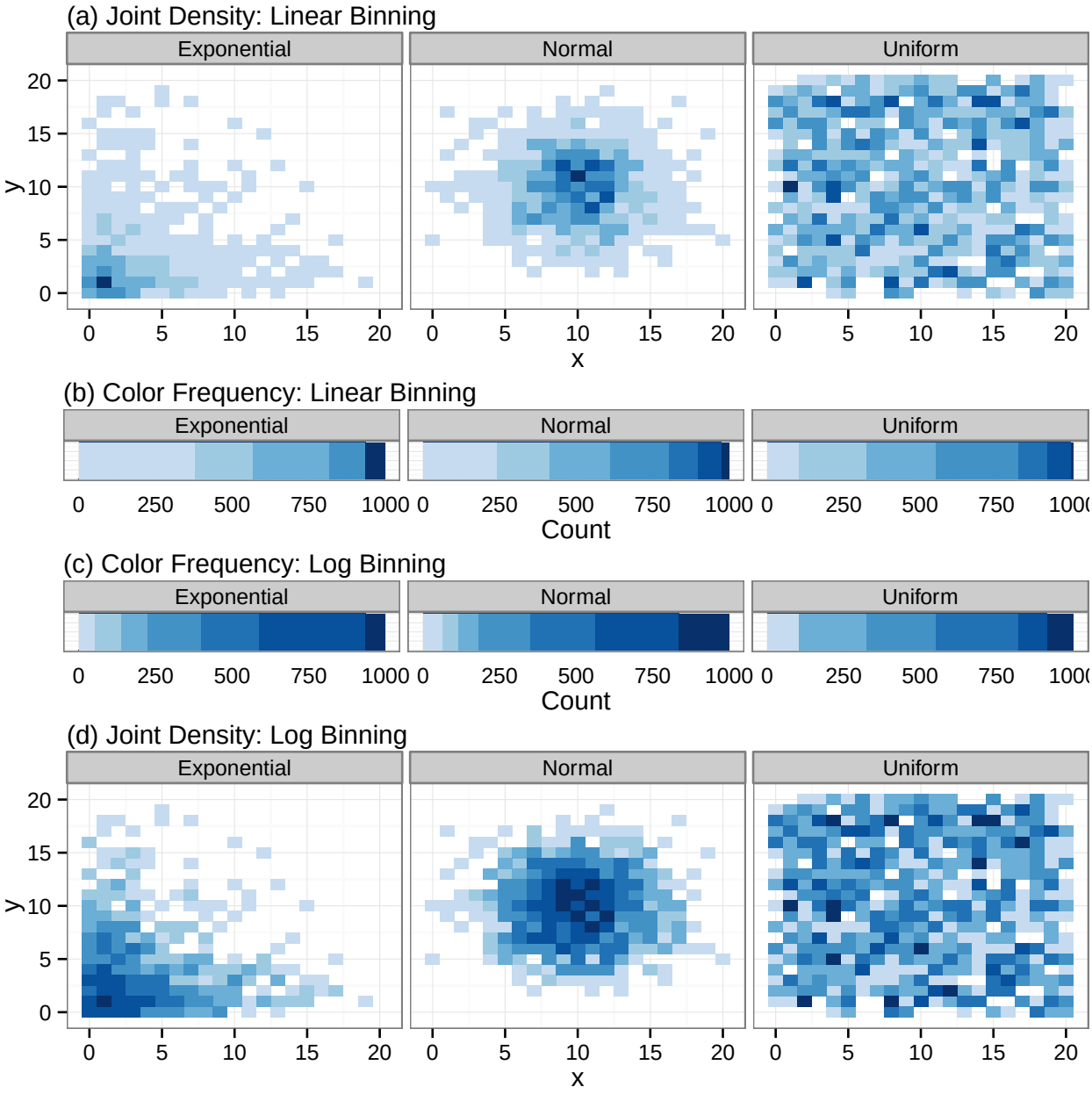


Figure 5: Linear and Log Frequency Binning to Simulated Data from Uniform, Normal, and Exponential distributions

explicitly shading bins according to frequency, more information is preserved than in a traditional scatter plot, as the frequency domain provides visual weight to tiles which represent more points. This generalization allows us to describe the plots in Figure 1d and e under the same framework as plots a-c.

By additionally increasing the bin width, we provide increasingly higher-level summaries of the data by smoothing over local structures. For small bin widths these structures are likely to be *noise* inherent in any real data set, while removing structures associated with larger bin widths likely masks real signal in the data.

4.3 Recovery of Numerical Information

As shown before, the overall loss partitions naturally into visual and numerical loss (see equation 2). We can make use of this and recover some of the numerical information lost due to binning by storing the numerical center $(\bar{x}, \bar{y})$ instead of the visual center (x^*, y^*) for use in subsequent calculations. Storing the binned data as the arithmetic mean rather than the visual center reduces the total loss, and the visual center can be recovered from the bin width and bin center (O_X, O_Y), see table 1 for a numeric example for this.

x	y	c
-0.2582	1.5497	1
-0.3843	5.6508	1
1.0597	2.4308	1
2.2680	1.5877	1
2.4992	4.6260	1
1.6788	5.0260	1
1.6054	6.4125	1
2.4180	6.4438	1
2.4345	6.4462	1
2.3620	10.3119	1

(a) Original Data, 100 rows

x^*	y^*	c
0	2	1
0	6	1
1	2	1
2	2	1
2	5	2
2	6	3
2	10	1

(b) Binned Data Visual Centers, 49 rows

$\bar{x}$	$\bar{y}$	c
-0.3843	5.6508	1
-0.2582	1.5497	1
1.0597	2.4308	1
2.2680	1.5877	1
2.0890	4.8260	2
2.1526	6.4341	3
2.2680	1.5877	1

(c) Binned Data Numerical Centers, 49 rows

Table 1: ten rows from Original and Binned Data Tables, with data storage sizes

Information recovery comes with a two-fold cost: the additional storage of numeric bin centers triples the amount of information stored, and the computational cost to compute numeric bin centers is $\mathrm{O}(nm^2)$, where n is the number of data points and m^2, the total number of bins. $\mathrm{O}(nm^2) \approx \mathrm{O}(n)$, as m is generally negligible compared to n.

In spite of the costs, information recovery is still useful for very large data, where the unbinned data set is computationally intractable. Minimizing numerical loss in this manner provides for more accurate calculations from the data, while maintaining the storage space advantages of binning.

5 Discussion and Examples

5.1 Comparison of Loss in Binning Algorithms

In order to quantify the advantages of each algorithm, we first compare the three sources of loss at different bin widths. The numerical and visual losses inherent in creating a binned scatterplot are a result of our desire to reduce and summarize the data until it is more manageable in size, and hence is a factor in our decision to bin the data in the first place. The random algorithm has higher numeric and visual loss than the standard algorithm, but in many cases the binned scatterplots are visually indistinguishable.

The numerical loss is a function of the distribution and bin width, and once the bin width is chosen, it is entirely fixed. The random binning algorithm will inherently have a larger numerical loss than the standard algorithm because it allows for random assignment of points to numeric bin centers further from the original data than does the standard algorithm.

As the visual loss is the difference between the numerical center and visual center of the bin, it is a function of the distribution and bin width (and the specific random assignment, in the case of the random algorithm). We would not expect to notice the difference in the visual loss in either algorithm, so long as the bin width is constant, because the expected value of the random process that results in shifting numerical bin centers is zero.

The far more important factor is the frequency loss, as we perceive that loss in the variability of the shading of bins which has a larger effect on the utility of the graphical data summary. The standard algorithm is preferable because it has lower total combined numerical and visual loss, but the two algorithms are very similar when we compare frequency loss. The random algorithm does have more variability, but the frequency loss values are much more similar, and when bins are of size two in either or both dimensions, the random algorithm has lower frequency loss than the standard algorithm. This holds for bins which are powers of two, as long as the random binning algorithm is applied iteratively to take advantage of the constant loss property noted above.

Investigation of the loss sources we might be tempted to conclude that the standard binning algorithm is superior due to lower loss, however the random binning algorithm displays strong advantages when encountering problematic data structure issues. Random binning is especially advantagous when discrete data values are recorded at a consistant increments but we bin our data using a binwidth that is a non-integer multiple of the incremental widths. In this senario, we see that the non-synchronous data increments and bin widths will cause bins to contain unconsistant numbers of possible data value. Assuming that data values are spread uniformly over the discrete values we want a visualization that displays uniform frequency. Using the standard binning algorithm the misalignment of data increment and bin widths will lead to and artifical frequency discrepancy that manifests in a striped pattern in the visualization. The random binning algorithm is expected to display the intended uniform frequency pattern because the randomization of data values to bin centers mitigates the problematic bin width selection.

5.2 Binning Loss in Baseball Data: Strikeout and Game Counts

For this data example we will revisit the baseball data used earlier in this paper. The characteristics of loss described in the discussion above are exemplified by viewing the loss for the baseball data in Figure 6.

Loss table for baseball example moved from beginning, this should not be its final resting place

We may also use this example to highlight the advantages of random binning. The two binned scatterplots at the top of Figure 7 show bins of the same width, but clear differences between the binning results are evident: while the standard binning algorithm has rounding artifacts that are clearly visible for even bin widths, the random algorithm produces a graph that better represents the true shape of the data. This is because the In contrast, when the bin widths are odd, the two binning algorithms yield nearly indistinguishable binned scatterplots, as shown in the bottom row of Figure 7.

5.3 Big Data: Airline Departure Times

The Federal Aviation Association (FAA) requires all airlines based in the United States to report details for every single flight. These are published online by the Bureau of Transportation Services at `http://www.transtats.bts.gov/DataIndex.asp`. Every day there are about 16,000 flights across the United States adding up to almost 6 Million flights a year. Scheduled and actual departure times for all flights in 2011 make up –in uncompressed form– a file of about 450 MB. A comparison of scheduled and actual departure times allows us an investigation of on-time performance of air carriers.

Figure 8 shows examples of two scatterplots of scheduled versus actual departure times. The plot on the left shows a sample of one million of those records. Even while using alpha blending this results in a severely over-plotted graph. On the right is a binned scatterplot of the complete data binned at 1-minute intervals. This does not have any spatial loss, as the times are recorded by minute. The 1-minute binning also reduces the data file to 176,384 individual records, less than 3% of the original data. Both plots show the same big picture patterns: scheduled and actual arrival times are highly correlated, recognizable from the conglomeration of points along the line of identity. Scheduled departure times past 6 am in the morning are much more common than earlier flights. It is much more likely for a flight to be delayed than to leave early, leading to the wash-out effect above the line, that is getting thinner with increasing delays. The range of delays on usual days starts at about one hour at 6 am and increases during the day to about 2 hours. The few number of flights before 6 am are also visible in both plots. The triangle of observations on the bottom right visible in both plots is nothing but an artifact of the data collection consisting of flights that are scheduled before midnight, but are delayed to departures past midnight. The cloud of outliers halfway between the two main structures is potentially interesting, since no immediate explanation comes to mind, and would be worthy of a follow-up investigation. What is not apparent in the plot on the left, is some fine-level structure that the plot based on all of the data shows. A close inspection of the plot on the right hand side reveals darker colored vertical lines at

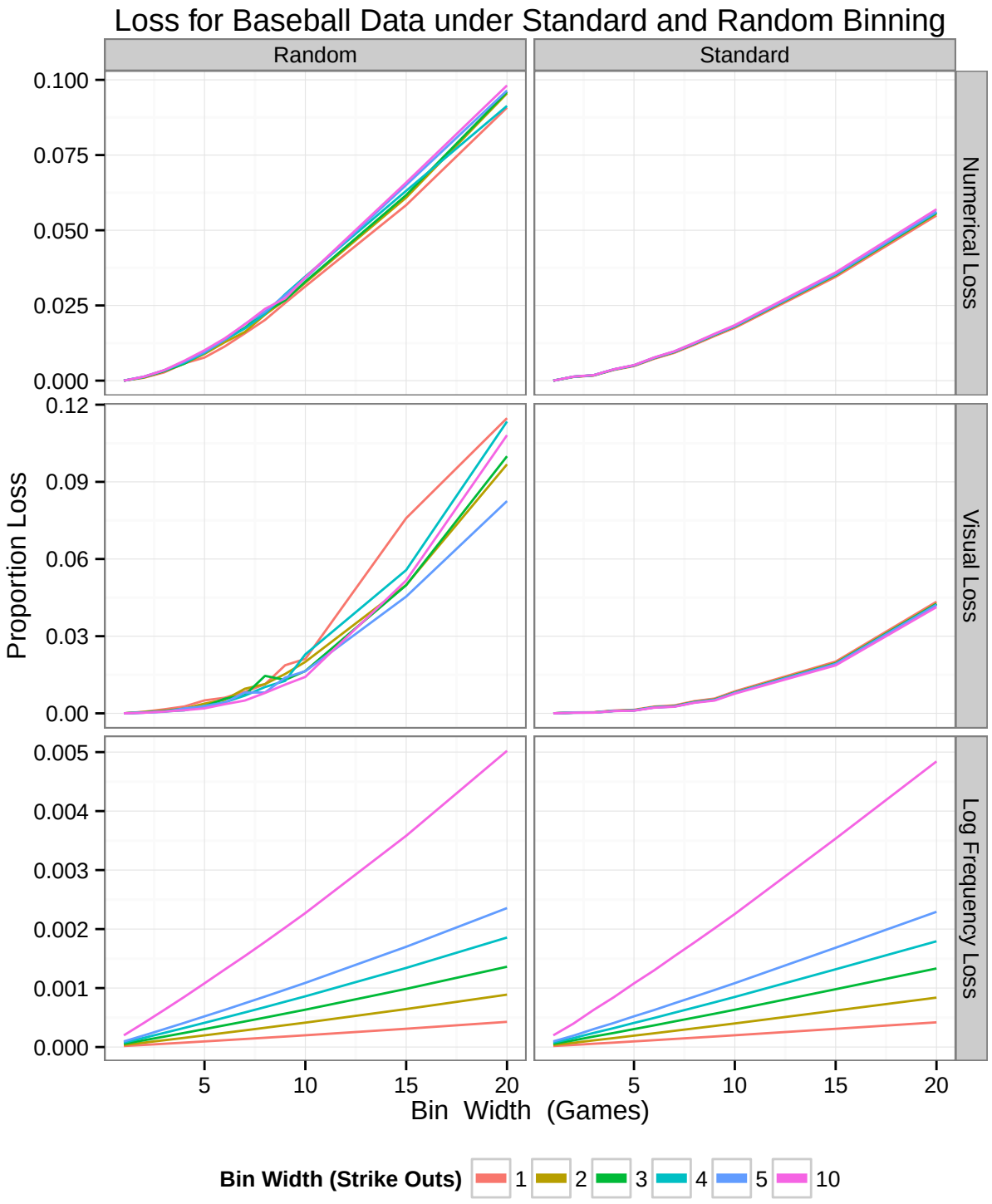
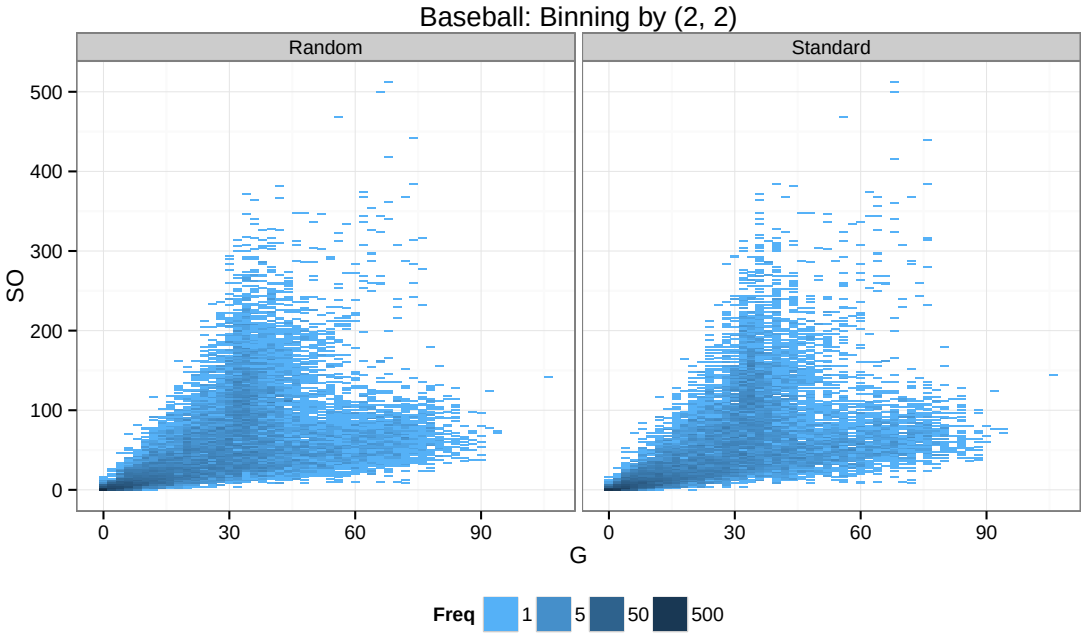
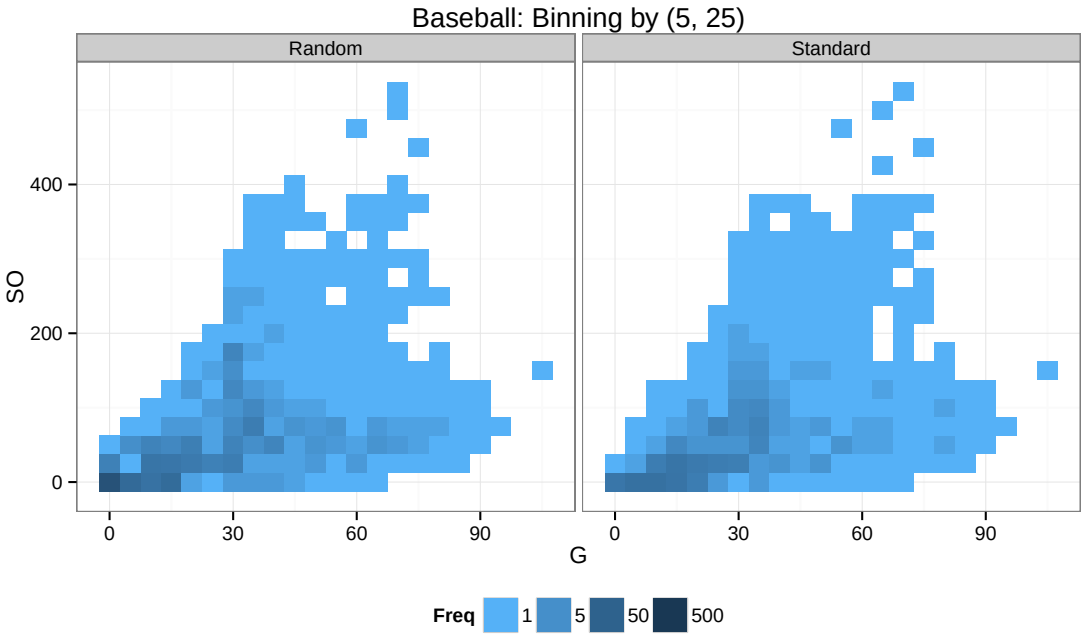


Figure 6: Loss funcion



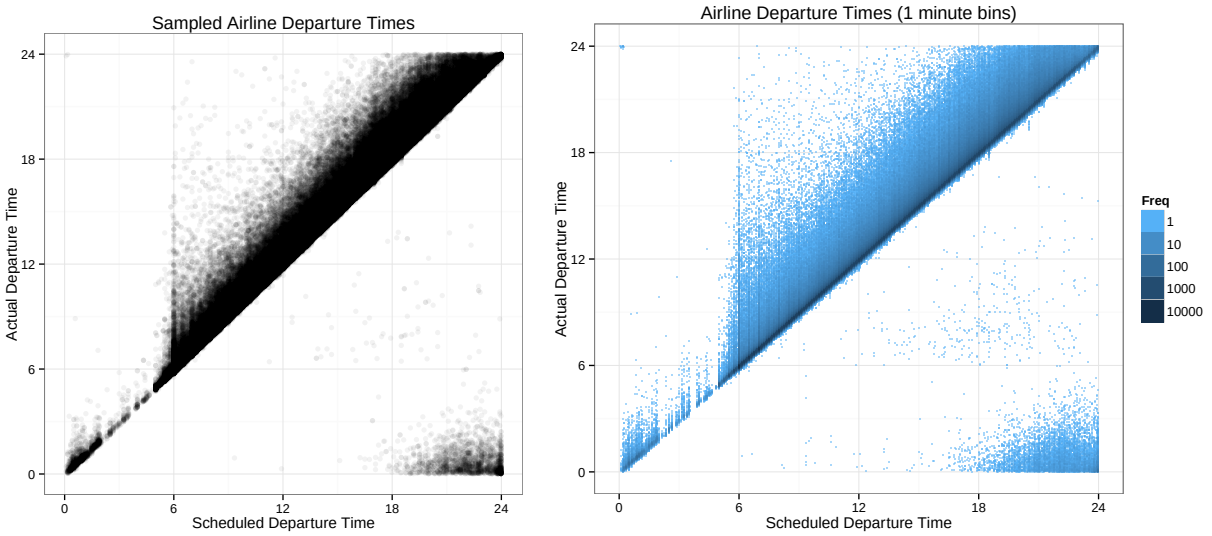
(a) (2, 2) tiling of G, SO data



(b) (5, 25) tiling of G, SO data

Figure 7: Comparison of random and standard binning: standard binning introduces artificial striping when the chose bin width is not a multiple of the data resolution.

30 minute intervals. It is obvious that more flights are scheduled with departures on the hour and at 30 minutes past the hour.



(a) Million-point sample of Airline Data

(b) Minimal binning of Airline Data

Figure 8: Scheduled and actual departure times of flights across the United States in 2011. The plot on the left is based on a sample of the data, the plot on the right shows all flights. The big patterns are visible in both plots, but the plot on the left misses some of the finer level details in scheduling that is visible in the plot on the right.

Binning data by five-minute intervals produces a more high-level summary of the relationship between actual and scheduled departure time, though it necessarily obscures some of the finer details. In addition, binning data by five minute intervals reduces the size of the data set to a much more manageable 19,787 observations which can be easily manipulated on probably any modern computer. Binning by 15-minute intervals reduces the data set to a nearly-trivial 3,575 observations, but the graphical summary becomes granular and less appealing at that resolution.

6 Conclusion

Large Data sets of continuous variables are very difficult to visualize in raw form, due to overplotting of points. Binning allows for the visualization and manipulation of large data sets, and easily translates into binned scatterplots which are more appropriate for the human visual system. The loss due to binning is the cost of reducing the data set to a more manageable size, but some of that loss can be recovered with additional computational investment.

We have presented two algorithms for binning; the standard algorithm, which has rounding artifacts but generally lower loss, and the random algorithm, which is a fast version of

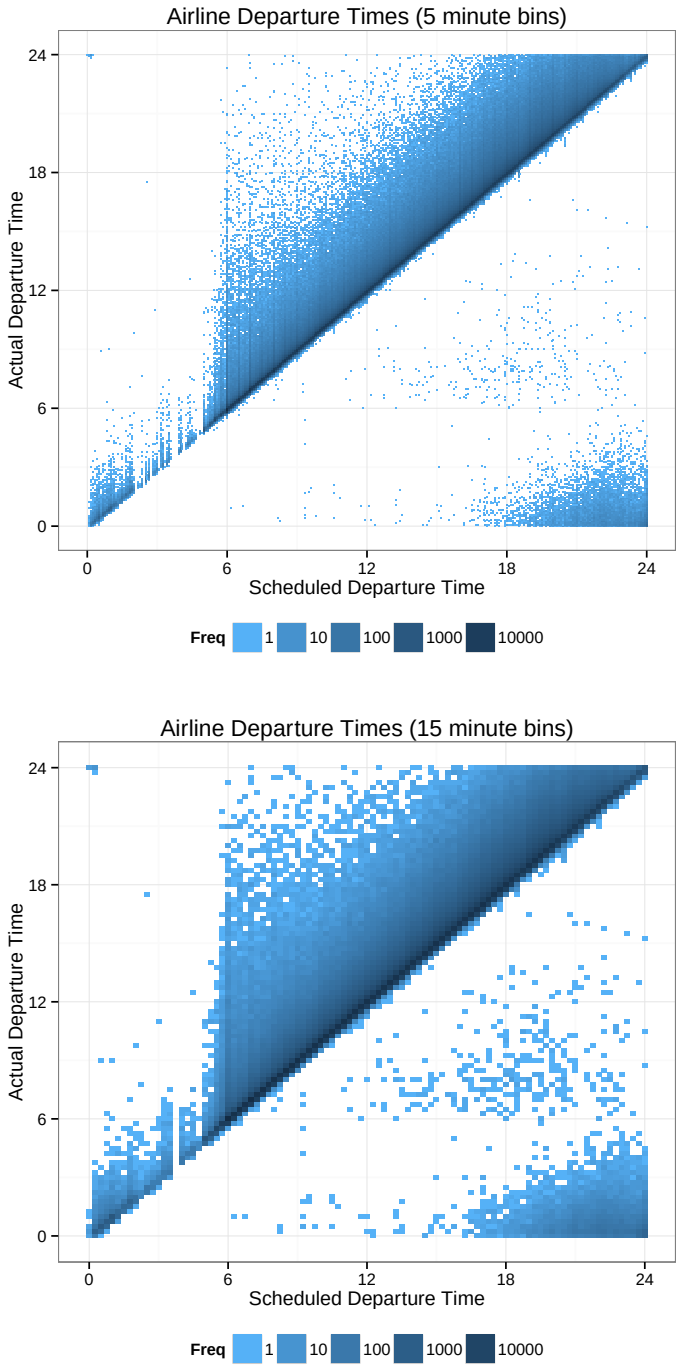


Figure 9: 5-minute bins produce a higher-level summary of the data than shown in Figure 8b. 15-minute bins produce an even more coarse summary of the data.

linear binning and is somewhat more computationally efficient. The choice of algorithm is a trade-off between speed and accuracy, and may be highly data set dependent. It is possible that there are computational modifications which could improve the accuracy of the random algorithm, but these modifications are likely to require additional computational operations.

A Mathematical Considerations

A.1 Partition of Spatial Loss

We want to show that spatial loss can be written as a partition of the form

$$L_s(x) = 1/S_\emptyset^2 \sum_i S_i^2 = 1/S_\emptyset^2 \left[\sum_i L_i^N + \sum_i L_i^V \right]$$

For that let us consider the values $\{x_i : b(x_i) = x^*\}$ in a single bin with center x^*. The numerical center of these points is given as $1/n \sum x_i = \overline{x}$. Then the total loss for each element x_i in this bin is:

$$\begin{aligned} S_i^2 = \sum (x_i - x_i^*)^2 &= \sum (x_i - \overline{x} + \overline{x} - x_i^*)^2 \\ &= \sum (x_i - \overline{x})^2 + 2 \sum (x_i - \overline{x})(\overline{x} - x_i^*) + \sum (\overline{x} - x_i^*)^2 \\ &= \sum (x_i - \overline{x})^2 + 2 \sum (0)(\overline{x} - x_i^*) + \sum (\overline{x} - x_i^*)^2 \\ &= \sum (x_i - \overline{x})^2 + \sum (\overline{x} - x_i^*)^2 \\ &= L_i^N + L_i^V \end{aligned}$$

Adding over all elements in the bin and all bins gives the formula above.

A.2 Constancy of Loss under Binary Splits

We want to show that under random binning the loss due to doubling of bin width is deterministic.

Let us consider the contribution L_i of a single point x_i to the loss when starting with the minimal bin width, i.e. each unique point is in a separate bin: doubling the bin width leads to two scenarios: a point is either at the bin center or directly between two bins. In the case that the point is at the bin center, its contribution to the total overall loss is zero, $L_i = 0$, and the associated probability $p_i = 1$.

Alternately, when the point is exactly half-way between two bin centers, the point is attributed with probability $p_i = \frac{1}{2}$ to either bin, leading to a loss of $L_i = \frac{1}{2} c_i$ regardless of which bin is assigned. Hence, in this case, loss is also entirely independent of which bin is assigned.

References

Carr, D. B., Littlefield, R. J., Nicholson, W. L., and Littlefield, J. S. (1987), "Scatterplot Matrix Techniques for Large N," *Journal of the American Statistical Association*.

Cleveland, W. S. and McGill, R. (1984), "The Many Faces of a Scatterplot," *Journal of the American Statistical Association*.

Friedman, J. H. (1997), "Data Mining and Statistics: What's the Connection?" .

Goldstein, E. B. (2007), *Sensation & Perception*, Belmont, CA: Thomason Wadsworth.

Healey, C. G. and Enns, J. T. (1999), "Large Datasets at a glance: combining textures and colors in scientific visualization," *IEEE Transactions on Visualization and Computer Graphics*.

Playfair, W. (1786), *Commercial and Political Atlas*, London.

Playfair, W., Wainer, H., and Spence, I. (2005), *Playfair's Commercial and Political Atlas and Statistical Breviary*, Cambridge University Press.

Theus, M. (2006), "Scaling Up Graphics," in *Graphics of Large Datasets*, New York: Springer.

Tukey, J. W. (1977), *Exploratory Data Analysis*, Lebanon, IN: Addison Wesley.

Unwin, A., Theus, M., and Hofmann, H. (2006), *Graphics of Large Datasets*, New York: Springer.