Visualizing Changing Probability Distributions

Björn-Aljoscha Kullmann
kullmann@in.tum.de

Supervisors: Paul Emmerich and Sebastian Gallenmüller
Seminar Future Internet WS 2016/17
Chair of Network Architectures and Services
Department of Informatics, Technical University of Munich

ABSTRACT

The visual representation of data is a powerful tool to aid scientific research and publications. Graphics can be used to illustrate the findings in an accessible way and help explore the meaning of experiments by displaying the data intuitively. Data often takes the form of probability distributions, such as the latency distribution of network traffic. These results can be hard to interpret for the human brain. Thus it is exceedingly important to aid the viewer with good graphical displays.

This paper gives an overview of how informative visualization can be achieved and how the data can unfold its story. Possible ways to deceive the viewer are explored and guidelines how to stay true to the data are given. Based on this knowledge, means to visualize individual data sets as well as series of experiment data are looked at. Two data series from the realm of network traffic measurement serve as example data for the graphics.

Keywords

Statistics, Graphical Display, Visualization

1. INTRODUCTION

The graphical representation is of great importance to every publication. Researchers rely on visual data representation to convey relevant information in publications. Especially statistical data is hard to understand without the right means to visualize it. This paper handles the central challenges when graphing probability distributions and highlights good ways to present statistical data accordingly. Two data sets are used to create the graphics for this paper. The realtime versus background traffic data set measures latency times in a stress test for high-speed network devices. The switch under test is tasked to prioritize forwarding the packets of the realtime traffic flow . The FLOWer concept utilizes the MoonGen packet generator and OpenFlow programmable switches to achieve high network traffic with customizable packets [5]. MoonGen is presented in the paper that is also the source of the other test data set [6]. Complex traffic patterns are generated by the novel packet generator to stress a software switch. The patterns looked at in this paper are a constant bitrate (CBR) flow and packets generated by a Poisson process [6].

The graphing tool used for the displays is the Python library matplotlib [10] and Seaborn [23], composed in Jupyter Notebook [15]. The code for the graphics is available in the LRZ GitLab repository [12].

The first chapter deals with the basic principles of visual presentation. The foundations of graphical perception are discussed and guidelines how to avoid pitfalls in the presentation of data. This is followed by various examples of how to present univariate data. The cumulative plot, dot plot and different kinds of histograms are explained. The sections dealing with the box plot and the violin plot show how summary indicators and distribution estimates can act in combination to produce a sound representation of the data. Finally, we take the step to visualize changing probability distributions over the course of a series of measurements. Utilizing advanced data representation techniques like graphic matrices, summary statistics and color makes the graphic overstep the two dimensions of paper to showcase multivariate data sets.

2. EXPRESSIVE DATA VISUALIZATIONS

2.1 Motivation behind Graphics: Presentation and Exploration

The most general term to describe data recorded in a human-readable fashion is the chart. Charts can take the form of a table to show numbers directly. If the data sets become too large to grasp from text alone or when relations in the data should be highlighted, a graphical chart should be chosen. Common charting tool like Microsoft's spreadsheet software Excel offer a preset number of chart topologies. Pie charts, bar charts or line charts can be created by a simple click. This might push users to present their data in a carelessly constructed display. A good graphing tool should instead provide users with means to implement their vision of a good display and help reveal the information hidden in the data [25]. Examples for powerful graphing tools are the programing language R [16] and the Python library matplotlib [10], the latter of which is used for the illustrations in this paper.

The first step when working with data should be to explore its meaning. Simple plots of the values from each batch or test series can give a first impression of the underlying distributions and relations. On that basis, more complex exploratory plots can be created to explore the facets and characteristics of the data. Once a thorough analysis of the data has been conducted, the results should be condensed into a well crafted presentation graphic. All conclusions made by the analyst should be easily reconstructible by looking at the graphic. Complete definitions and explanations are important to support full comprehension of the data [22].

doi: 10.2313/NET-2017-05-1_02

2.2 Scale and Comparability: Adapting data to a frame

To bring numbers into a graphic, we have to define the space they should exist in. The graphic frame is defined by its axis. On a two dimensional surface of a book page, these are formed by the horizontal x-axis and the vertical y-axis. The scales measure the contents of a frame along their axis. Scales are driven by the data. With nominal scales, data measures falling into different categories can be differentiated. These categories have no inherent ordering. An arbitrary order can be chosen as appropriate and categories are usually spaced out evenly along the axis. An ordinal scale enforces total ordering over all possible values. The value marks should be drawn on the axis in their ascending order. An interval scale can be applied when a range of values is looked at. Adding one value positioned at a point on the axis to another value, the resulting value will be placed at the position equal to the sum of the added values' distance to the axis origin. A ratio scale demands such behavior for magnitude ratio comparisons between values [25].

Multiple variables can be plotted on the same axis. They have to share the same unit and magnitude level. The International System of Units (SI) describes base classes and transformation rules for units like length, weight, time or temperature. From these base classes, composite measurements can be derived to represent an interplay of SI units, for example volume, pressure or power. Basic categorical dimensions and scalar values are not part of the SI system and are therefore called dimensionless measurements [25].

Choosing the scale greatly influences the perception of the data. It can be a good idea to choose the axes whose boundaries extend slightly beyond the minimum and maximum of the data. Including zero is beneficial, because it serves as a good baseline and point of orientation for the viewer. If the data demands an origin different from zero, the reasoning behind the new baseline should be made clear. If multiple graphics show similar data it can be wise to choose the same scale to facilitate comparison between them [22]. The frame lines can be adjusted to reflect properties of the data. Cropping the frame lines to extend only to the maximum and minimum values produces a so called range frame. Ticks along the frame line can not only be used to present a regular spacing of the data, but also to mark single values or special properties of the data, like quartiles [20]. Reporting too much information through the frame however might clutter the display and confuse the viewer [22].

All data should fit nicely into the frame. Choosing an oversized scale to incorporate large values can obscure small features of the distribution. Splitting up the graphic into multiple figures to show each cluster individually should be considered. An alternative approach would be to re-express the scale [21]. Moving from a linear to a logarithmic scale can be a good choice for rapidly growing data. Thereby, multiplicative distances between data points turn to additive and ratios to differences. The most common log base is 10, but the base most appropriate for the data should be chosen. Scale breaks may be chosen to avoid wasting blank space between values. Two tilted lines breaking a scale line should be used to indicate a partial scale break, the line resumes with at a higher value. Values must not be connected across scale breaks. A full scale break can even change the resolution of the scale. A full vertical line at each of the breaking ends should indicate a shift in scale [2].

When all the scaling work is done, the shape of the graphic itself has to be chosen. The proportions of the display should orientate themselves in relation to the data. Regression displays might benefit from a square frame, because the graphic is split in two equally large sections along the 45° line. Vertical displays imply stark growth, while horizontal frames facilitate the impression of change over time. In general, it is more pleasant to watch a graphic wider than tall. The proportions can be chosen accordingly. Following the Classical ideal of ancient Greece, aesthetic can be determined by the golden section with height $a = 1$ and the width $b = \frac{\sqrt{5}+1}{2} \approx 1.618$ [20]. The common aspect ratios in media can be a reference, too. The ratio $4 : 3 \approx 1.333$ leans more towards the square. The modern standard TV resolution $16 : 9 \approx 1.778$ is closer to the golden section. The cinema ration $19 : 10 = 1.9$ is almost twice as wide as it is tall. The graphics in this paper will use the $16 : 10 = 1.6$ ratio, because it is the standard that comes closest to the golden section.

2.3 Graphical Perception

The effect of a visual display is determined by the viewers perception of its content. Great data sets have no merit when the observer cannot make out its meaning. Values should be presented so that they can be judged effectively. Weber's law suggests that the contrast in magnitude of two physical values is not perceived by their absolute difference. Instead, the human brain distinguishes their ratio. Two line segments A and B that are not aligned are read as "line A is 30 percent longer than line B" instead of "the difference in length between between A and B is 2 centimeters". Therefore displaying values that differ in ratio is important when they do not align to a common base [2].

Steven's law establishes a metric for how judgment of magnitudes differ from reality. The perceived scale $p(x) = cx^{\beta}$ is skewed by a constant c and a power β to the magnitude x. When testing for the judgment of lengths, mild error factors β of 1 ± 0.1 have been estimated. For area, this error becomes more severe to levels of 0.6 to 0.9. It only gets worse for volume with average β of 0.5 to 0.8. Ratio judgment therefore become increasingly skewed for unintuitive attributes [2].

Cleveland assigns graphical elements to a hierarchy of graphical perception tasks [2]. Graphical elements that can be judged more easily should take precedence in graphical design and elements further down the hierarchy should only be chosen once the limits of better attributes have been reached. The hierarchy is divided into 7 levels. Attributes in the same level do not differ in their quality of magnitude perception.

1. Position along a common scale and axis
2. Position along a identical scales with nonaligned axes
3. Length
4. Angle and Slope
5. Area
6. Volume
7. Color hue, Color saturation, Density of occurrence

doi: 10.2313/NET-2017-05-1_02

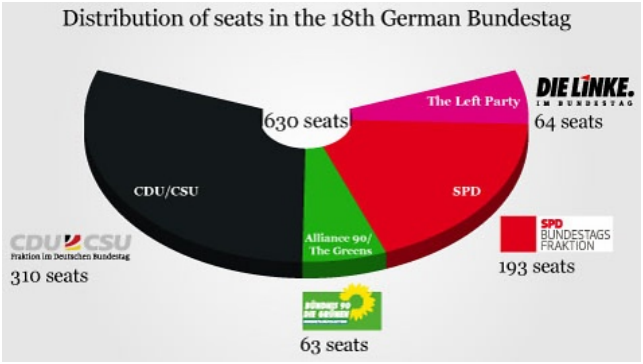

Figure 1: Party seat distribution of the 18th German parliament [3].

Tufte introduced the Lie Factor [20] to express how the physical measure on the graphic surface relates to the numerical quantities represented.

$$\text{Lie Factor} = \frac{\text{size of effect shown in graphic}}{\text{size of effect in data}} \tag{1}$$

A lie is induced by a misrepresentation of quantities. A linear value that is represented by an area has exaggerated graphical impact in relation to the more fitting line or dot along its axis. Even bigger perceived lies can be told when expressing a one dimensional value by volume portrayed through an additional foreshortened axis. In consequence, values become hard to decipher and heavily skewed, even before perceptive error comes into play. The principle to avoid this kind of lie is to have the number of informative dimensions depicted not exceed the number of dimensions in the data [20]. Every graphic property not bound to a data value introduces the potential of misjudgment.
A value is well represented by an area if it is naturally computed by the product of two values or the the integral over a line [20]. Compare the histogram to a bar chart. In the bar chart, the bars extend to a single value and are merely a figurative representation of a dot value. The histogram, described in later chapters, on the other hand groups observations over the bin range on the x-axis and maps their count to the y-axis [25].
A similar case can be made for pie charts. The viewer is challenged to compare angle and radian ratios combined with narrow areas [20]. Data that actually relies on polar coordinates can be visualized with a rose diagram. This type of graphic is inspired by the wind rose that has been used for centuries on navigational charts. It is especially useful for data distributed around compass points, for example wind direction data [25]. An exemption from this principle could be made to represent data in a familiar real world context [25]. The distribution of seats resulting from the parliamentary elections are often presented in a manner resembling the parliament hall. Figure 1 shows the distribution of the party seats along the traditional political spectrum from left to right (in this case from "behind", center-right to left), as they might actually be located in a sitting of parliament [3].

For the construction of graphics, less is often more. Erasing as much non-information carrying elements as possible makes a graphic more concise. The data-ink ratio measures the portion of ink that is actually devoted to showing numbers. All ink in the graphic should add information to the graphic that was not there before. This holds especially true for fancy decoration, extensive grid lines, frames and grid ticks [20]. Graphics today are mostly generated by computers and the underlying data is often publicly available as precise digital records. Grid lines lose importance for plotting and retracing exact data values. It becomes more essential to show the inherent information than the exact values [22]. Tick marks and labels can instead be used to mark important events in the data.
In conclusion, graphics should always meet their purpose to convey new interesting information. Small and highly labeled data sets are often better suited for tables and text. The graphic designer should bear the basic principles of graphical perception in mind. Only when information is accurately conveyed, the design of the data graphic is a success.

2.4 Kernel Density Estimate

The key challenge when looking at statistical data is to know the underlying probability distribution. One way to estimate the distribution function of the data is to compute the kernel density. The result is a continuous approximation of the probability function, prominently featured by the violin plot we expand on later. The kernel is a tool to assess the target probability function by weighing the samples of the data. It is formed by a probability measure that should be similar to the target probability function. In practice, standard kernel like the ones described below yield good results even for complex distributions. By taking a sample of n values x_i from the data set, the density function estimate can be deduced by aggregating the kernel $K(x)$ for each of the values x_i [14]:

$$\hat{f}(x_0) = \frac{1}{n}\sum_{i=1}^{n} K_h(x_0 - x_i) \tag{2}$$

A commonly used kernel is the the Epanechnikov kernel, a truncated parabola

$$K_E(t) = \frac{3}{4}(1-t^2) \quad -1 \leq t \leq 1. \tag{3}$$

The standard normal kernel, also called Gaussian kernel [25], given by

$$K_N(t) = \Phi(t) = \frac{1}{\sqrt{2\pi}} e^{-0.5t^2}, \tag{4}$$

can be more convenient for its smoothness [14].
Additionally, the kernel function is weighted by a bandwidth factor h such that $K_h(t) = \frac{K(t/h)}{h}$. The bandwidth determines how strong the kernel for each kernel sample influences a point in the density function. This limits the spread of the kernel [25]. Choosing a large value for h leads to oversmoothing, a small h results in an unstable multimodal estimate [14]. Assuming a normal distribution for the data, an optimized bandwidth is achieved by the normal reference rule [18]:

$$h = 1.06\hat{\sigma}n^{-1/5}. \tag{5}$$

Thereby $\hat{\sigma}$ denotes the estimated standard deviation and can be deduced from the sample standard deviation (ssd) of the data. A more stable choice for data deviating from the assumed Gaussian distribution is the interquartile range

doi: 10.2313/NET-2017-05-1_02

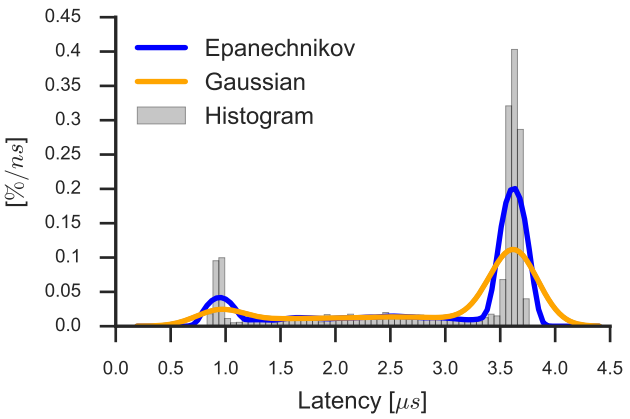

Figure 2: Kernel density estimates of priority traffic latency times with an Epanechnikov kernel and a Gaussian kernel. The bandwidth has been chosen by Scott's normal reference rule. A thinly binned histogram shows the distribution.

IQR of the data set: $\hat{\sigma} = min(ssd, IQR/1.34)$ [13]. The final kernel density function for a Gaussian kernel using the normal reference rule can then be computed:

$$\hat{f}(x_0) = \frac{1}{n}\sum_{i=1}^{n}\frac{e^{-((x_0-x_i)/h)}}{h}. \quad (6)$$

Figure 2 compares an Epanechnikov kernel and a Gaussian kernel estimate of a highly bimodal data set. Both estimates use Scott's normal reference rule for the bandwidth. The Epanechnikov kernel fits the peaks of the data more closely, while the Gaussian kernel applies stronger smoothing to the extreme values.

3. VISUALIZING UNIVARIATE DISTRIBUTIONS

3.1 Cumulative Plot

Measuring a statistical variable raises the question of its distribution. Discrete variables can easily be visualized by plotting their values against their respective absolute or relative frequency in the measurement. With continuous data this becomes impossible, since each observation might not be identical to any other observation. The frequency of individual continuous values is one and therefore insignificant in the overall data set. To visualize a continuous distribution we can use a cumulative plot. The graph lists the range of values on the x-axis. It starts from zero on the y-axis and each sample adds one to the vertical axis at the x-position of its value until all observations are processed. Figure 3 shows such a plot for the relay of prioritized traffic. Two slopes with an increased amount of arriving packages are visible at approximately $1\mu s$ and $3.7\mu s$ and a steady stream in between. Yet the display only shows the cumulated values and their intrinsic distribution is not obvious. A different approach is needed to show the distribution.

3.2 Dot Plot

The most simple solution to show a distribution is the dot plot. It shows the sample values on a one dimensional number line. It can be a good starting point to get to know a distribution [14]. The plot is only sensible for small data sets. Once the points start to overlap, it is possible to offset the point in the vertical axis [2], effectively forming a tally for a region of values on the number line corresponding to the size of the dot. Figure 4 shows such a stacked dot plot of a randomized sample of 100 data points in the realtime traffic data set. Note how the peaks of the bimodal distributions and the sparse values in between are clearly visible. Still, the display reaches its limit even for relatively few data points. Would we visualize more samples, the graphic would become convoluted and confusing.

3.3 Histograms

The concept of the histogram solves this problem. It is most commonly associated with the visualization of probability distributions widely used in publications from a wide range of scientific fields. This concept is also known by the term frequency diagram [1].
By dividing the range of the data values into intervals called bins, the number of observations falling into each bin become countable and bins can be plotted as bars [25]. For meaningful comparability of the bins, especially when bin sizes are different, it is important to note that the number of observations in a bin should be proportional to the area of the bin bar [1]. This ensures that larger bins are not excessively weighted in comparison to their smaller counterparts [14]. The vertical axis then shows the bin count divided by the bin width and the total number of observations. This denotes the average relative frequency for the bin.
The result is a density histogram, plotting the bin count v_j and the bin width h_j in the bin $B_j = [t_j, t_{j+1})$ that covers the interval between j and $j+1$, to the density estimate of each bin, relative to the sample size n [14]. The area of a bin then represents its bin count. The unit for the bin height is therefore in percent per width unit, in this case percent per nanosecond [8].

$$\hat{f}(x) = \frac{v_j}{nh_j} \quad x \in B_j. \quad (7)$$

Choosing an appropriate number of bins is the key step to make a histogram that reflects the data well. A rule of thump is to use $\sqrt{n}$ bins for n samples [24]. The bin width should furthermore respect a possible granularity of the data

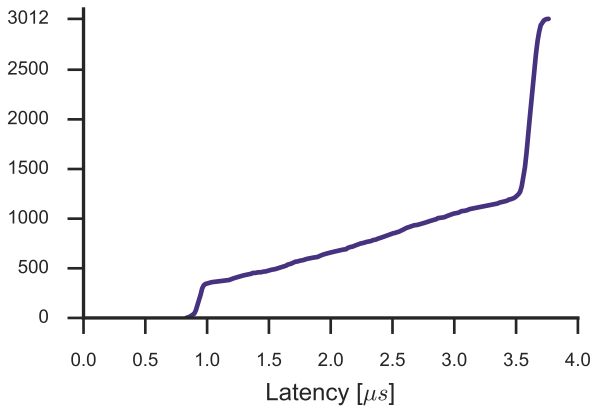

Figure 3: Cumulative plot of priority traffic latency times with 3012 observations.

doi: 10.2313/NET-2017-05-1_02

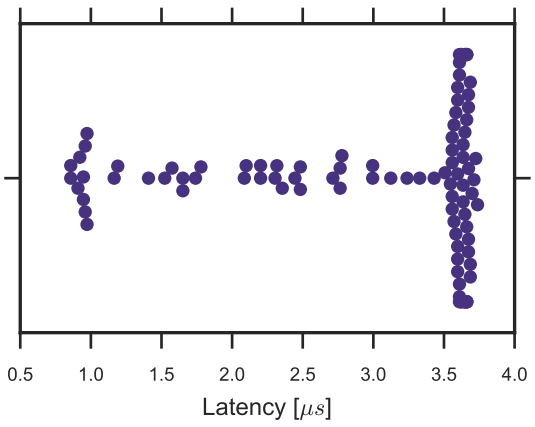

Figure 4: Dot plot of 8 Gbit/s realtime traffic with vertically offset dots to avoid overlapping.

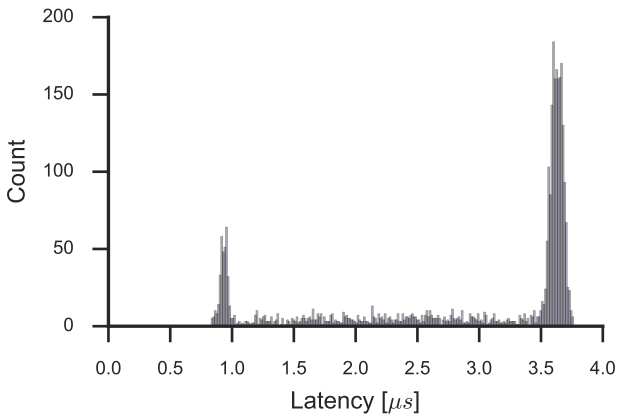

Figure 5: Histogram of priority traffic with a minimum bin size of $12.8ns$ determined by hardware granularity.

and should not be smaller than the smallest difference between two adjacent values [25]. Choosing the bin width as an integer multiple of the granularity can also add clarity to the display [26].
More sophisticated bin count estimates can be derived from the statistical attributes of the sample data. Sturges [19] deduces a good number of bins to be $k = \log_2 n$. This was further refined by Doane [4], who recommended $k = 3 + \log_{10} n \log_2 n$ to account for skewness [25]. To estimate a good bin width h of a bin, Scott [17] proposes the sample standard deviation ssd as the deciding factor to calculate $h_S = 3.5 ssd n^{-1/3}$. Similarly, Freedman and Diaconis [7] suggest using the more robust interquartile range IQR, resulting in about 30% lower bin width than Scott's for a normal density [14]:

$$h_{FD} = 2IQRn^{-1/3}. \tag{8}$$

The rules should only serve as a general guideline. For presentational graphics, the bin width should be chosen in a way that presents the data with a tolerable loss in accuracy [2].
The data for the example histograms is taken from the realtime and background traffic experiment. The data set has the following properties:

- Number of samples: 3012
- Minimum value: 832.0
- Maximum value: 3763.2
- Sample standard deviation: 997.62
- Interquartile range: $3635.2 - 2265.6 = 1369.6$
- Sample granularity: $12.8ns$

Figure 5 uses a number of bins that is determined by the granularity of the measuring hardware, resulting in 229 bins. The two peaks at the beginning and the end of the data range are clearly visible. The part in between is made up of a constant noise of packages coming through at other

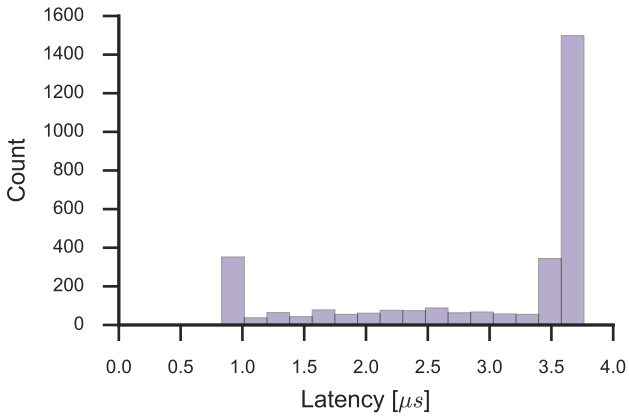

Figure 6: Absolute frequency histogram of priority traffic with the bin size determined by the Freedman-Diaconis rule.

latencies. The histogram of figure 6 applies the Freedman-Diaconis rule $h_{FD} = 2 \cdot 1369.6 \cdot 0,07 \sim 15.45$ rounded up to 16 bins. The bimodal distribution is still explicitly visible, but some of the local peaks in the data have been oversmoothed. The $\sqrt{n}$-rule results in a number of 55 bins in figure 7. Sturges' $\log_n$ only leads to 12 bins. Using the sample standard deviation according to Scott computes 255 bins, a result heavily skewed by the data's deviation from the normal distribution. A minimal histogram in figure 8 of 3 bins shows only the peaks and the noise in between with differing bin width. Grouping empty or sparsely occupied bins together can remove clutter from the display. When grouping bins together, it is important to keep the ratio between the bars intact and not mask data points that could be interesting to the reader.

3.4 Box Plots and Violin Plots

A schema [26] is a concise display that shows characteristic features of a distribution [2]. The box-and-whiskers-plot [21], or simply box plot, is the most common schematic plot, featuring the median, the first and third quartile, as well as a measure to identify values considered normal or

doi: 10.2313/NET-2017-05-1_02

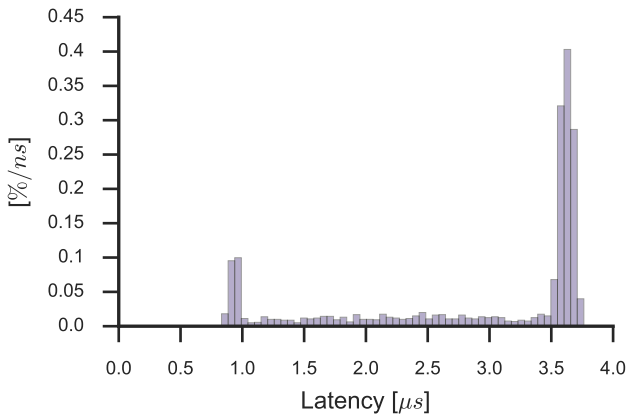

Figure 7: Density histogram of priority traffic with the bin size determined by the $\sqrt{n}$ rule.

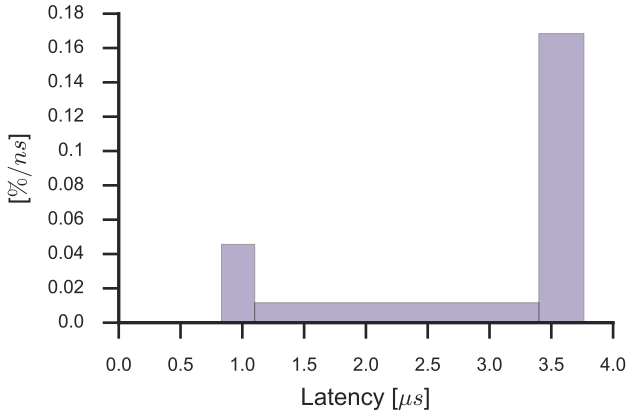

Figure 8: Density histogram of priority traffic with variable bin size to only show the peaks.

outliers.
The box plot was initially described by Tukey [21] and refined by Wilkinson [25]. The first step is to outline a thin box with its lower edge at the 25th quantile (1st quartile) and the upper edge at the 75th quantile (3rd quartile), forming the hinges of the box. The box is then crossed with a horizontal line at the position of the median.The whiskers stretch out from the hinges to the lower and upper fences of the plot. The position of the fences are determined by the interquartile range times one and half from the corresponding hinges: $whisker\_range = hinge \pm 1.5 \cdot IQR$. Values beyond these fences are treated as outliers and are each marked with a dot.
Figure 9 shows a box plot of constant bitrate traffic, which was generated by the MoonGen traffic generator. The median is closer to the top of the box, indicating a distribution skewed towards higher values. A single observation exceeds the normal range of the data with a latency of more than $140\mu s$.
The box plot is best suited for data with a steady unimodal distribution. Peculiarities such as bimodal distributions are masked by the terse display of summary statistics. The vio-

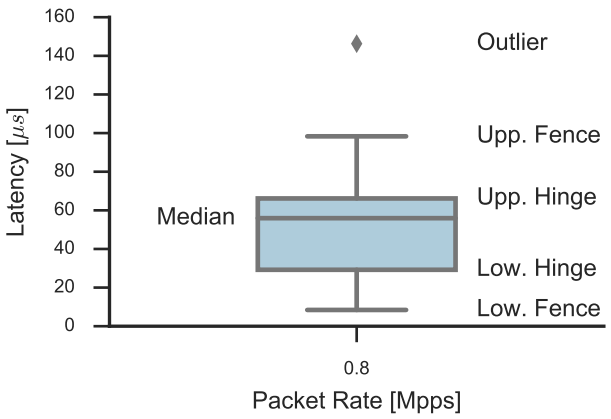

Figure 9: Box plot of latency times at $0.8 Mpps$ CBR traffic. The schematic features are labeled accordingly.

lin plot adds additional information to the box plot [9]. The box becomes a thick line and thinner lines extend to the positions of the upper and lower fences. A circle marks the position of the median. Then a kernel density estimate is plotted symmetrically to both sides of the box, resembling the form of a violin. Outliers are not tagged by any symbols. When comparing multiple violins, the scaling of the density estimate should be chosen according to the intended effect of the comparison. By scaling the violins to the same area or maximum width, the distributions can be compared effectively. Scaling proportional to the sample size places emphasis on the differences in data population.
The categorical nature of box and violin plot makes it possible to arrange the plots for multiple data series on one axis. A special variant of the violin plot is to make it asymmetrical [11]. Each side's density estimate shows a different subgroup of the data. This enables direct comparison of the two distributions.
Figure 10 shows the latency measurements for multiple packet rates in a single graphic. The kernel density graph for the CBR traffic, that is shown on the left of the box for the $0.8 Mpps$, reveals the deep valley that was hidden in the box plot visualization of figure 9. The density estimates are also easily comparable to the measurements for other packet rates. The bitrate CBR kernel density estimates form the left side of each violin and are opposed by the Poisson traffic type density estimates on the right side. A trend towards a more gradual slope for the Poisson latencies can be surmised.

4. MULTIVARIATE PLOTS

If we want to look at data varying in more than two dimensions, we have to find find strategies to make all the values accessible in the two-dimensional space of paper. When evaluating series of measurements, a so called small multiple display can help structure the data. The visualizations of each measurement are arranged as a matrix. Additional information can be encoded by using the vertical and horizontal axes to vary two different variables, or the series can be alloted arbitrarily for categorical or ordered from left to right and top to bottom for a single ordinal scale. It is especially important for this display that the design remains constant

doi: 10.2313/NET-2017-05-1_02

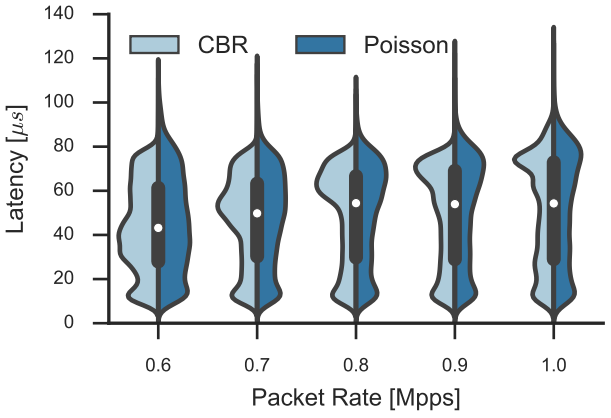

Figure 10: Split violin plot series of latency times from 0.6*Mpps* to 1.0*Mpps* CBR and Poisson traffic, scaled to the same area.

for all sub-graphics. All scales and color schemes have to be linked, enforcing an identical display for all frames [20]. Figure 11 shows a small multiple for the MoonGen data, comparing the constant bitrate against the Poisson process latencies of generated packets on the x-axis and measurements for different packet rates on the y-axis.

A simple way to reduce the dimensions of the data is to conflate each measurement to a numerical indicator. Summary statistics provide a simple solution for expressing a distribution in a single value. Taking the arithmetic mean or the interquartile range of a sample subset is inevitably associated with information loss, but trends in the data are still easily traceable. Figure 12 plots the medians of the CBR data and accompanies them with vertical lines corresponding to their interquartile range. This effectively produces a simplified box plot series.

Instead of introducing new dimensions in space, color can be used to represent a numerical magnitude. A heatmap features tiles arranged on the grid of a 2D graphic. Each tile is then colored by a third variable to reveal patterns in the distribution [25]. In color theory, color space can be modeled as a cube with the primary colors black, red, green and blue and the secondary colors white, magena, cyan and yellow serving as edges. These colors are the extreme coordinates and colors in between can be achieved by interpolating between the edges. To express a triple of data values, the numbers need to be normalized to fit inside the color cube and the result can be used as a color value to be painted on a graphic. Because color is likely perceived as ambiguous, graphics should resort to a single color dimension. Using brightness makes it possible to shade elements continuously, even without the need to print color. Hue is the spectral component of color, for example red, green or purple. Easily distinguishable color schemes are well suited to represent categorical data. Saturation describes the degree of pure color, or rather amount of hue, in a patch. This color dimension transits from fully saturated color to gray without changing the brightness. An easy distinction between different levels of saturation is not easily possible. An attribute like uncertainty might therefore be well suited to be expressed by saturation [25].

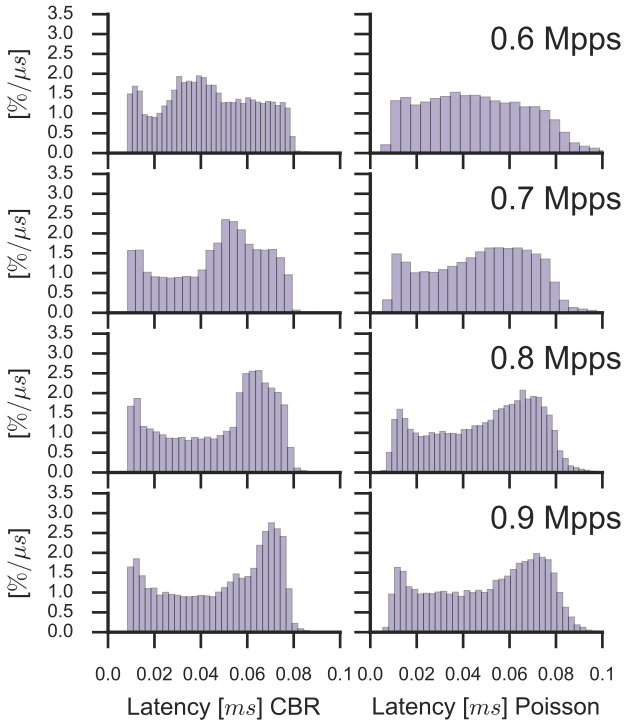

Figure 11: A small multiple comparing CBR and Poisson traffic over the course of four measurements with different packet rates, indicated in the top right of each row.

Figure 13 uses 55 bins to present the Poisson traffic data for different packet rates. The color reaches from a light blue for low frequencies to a dark blue for highly populated bins. Each packet rate row is normalized in order to give the rows with lower packet rates and fewer events the same visual impact as the rows with higher packet rate and therefore higher event density. The valleys and peaks are visible, but more smoothing might be necessary for the color histogram to become a sound data display.

5. CONCLUSION

Creating a good statistical graphic means to iterate many steps of graphical design, evaluation and redesign. When confronted with a probability distribution, basic plots like the dot plot and the histogram can give a good idea of the data. Bivariate data can be graphed on a scatterplot.

The second step is to take it further and highlight relations and peculiarities inherent in the data. Kernel density estimates and summary statistics help create concise comparisons of data subsets. A third variable dimension can be used by introducing color, in form of a heatmap or a color histogram. The analyst is encouraged to experiment creatively. The best way to present the data at hand might not be found in traditional charting conventions. A good graphing tool aids the analyst on the quest to find the perfect visualization.

References

[1] G. E. Box, J. S. Hunter, and W. G. Hunter. Statistics for experimenters. 2005.

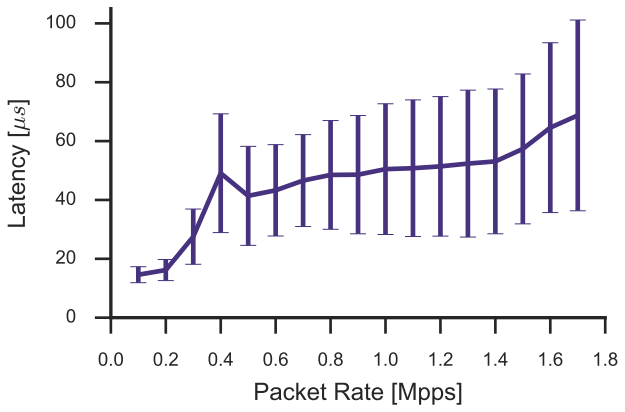

Figure 12: CBR traffic latency means with IQR ranges modelled as error bars.

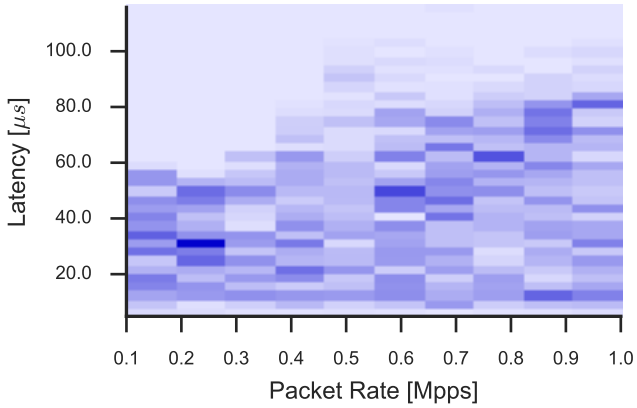

Figure 13: A color histogram of Poisson traffic for a series of tests with different packet rates. The more intense blue shows regions with higher frequency of occurrence.

[2] W. S. Cleveland et al. *The elements of graphing data*. Wadsworth Advanced Books and Software Monterey, 1985.

[3] Deutscher Bundestag. Sitzverteilung des 18. Deutschen Bundestages. `https://www.bundestag.de/bundestag/plenum/sitzverteilung_18wp`, 2015. Accessed on September 26th 2016.

[4] D. P. Doane. Aesthetic frequency classifications. *The American Statistician*, 30(4):181–183, 1976.

[5] P. Emmerich, S. Gallenmüller, and G. Carle. Flower – device benchmarking beyond 100 gbit/s. In *IFIP Networking 2016*, Vienna, Austria, 2016.

[6] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, and G. Carle. Moongen: A scriptable high-speed packet generator. In *Proceedings of the 2015 ACM Conference on Internet Measurement*, pages 275–287. ACM, 2015.

[7] D. Freedman and P. Diaconis. On the histogram as a density estimator: L_2 theory. *Probability theory and related fields*, 57(4):453–476, 1981.

[8] D. Freedman, R. Pisani, and R. Purves. Statistics. *New York*, 1980.

[9] J. L. Hintze and R. D. Nelson. Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998.

[10] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[11] P. Kampstra et al. Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of statistical software*, 28(1):1–9, 2008.

[12] B. Kullmann. Code for the Future Internet Seminar. `https://gitlab.lrz.de/kullmann/Future_Internet_Seminar`, 2016. Accessed on September 28th 2016.

[13] B. Liu, Y. Yang, G. I. Webb, and J. Boughton. A comparative study of bandwidth choice in kernel density estimation for naive bayesian classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 302–313. Springer, 2009.

[14] M. C. Minnotte, S. R. Sain, and D. Scott. Multivariate visualization by density estimation. In *Handbook of Data Visualization*, pages 389–413. Springer, 2008.

[15] F. Pérez and B. E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, 2007.

[16] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2008.

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

[18] D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 1992.

[19] H. A. Sturges. The choice of a class interval. *Journal of the American Statistical Association*, 21(153):65–66, 1926.

[20] E. R. Tufte and P. Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, 1983.

[21] J. W. Tukey. Exploratory data analysis. 1977.

[22] A. Unwin. Good graphics? In *Handbook of data visualization*, pages 57–78. Springer, 2008.

[23] M. Waskom et al. seaborn: v0.7.1. `https://doi.org/10.5281/zenodo.54844`, 2016.

[24] M. B. Wilk and R. Gnanadesikan. Probability plotting methods for the analysis of data. *Biometrika*, 55(1):1–17, 1968.

[25] L. Wilkinson. *The grammar of graphics*. Springer Science & Business Media, 2006.

[26] G. Wills. *Visualizing time: Designing graphical representations for statistical data*. Springer Science & Business Media, 2011.