Department: Visualization Viewpoints
Editor: Theresa-Marie Rhyne, theresamarierhyne@gmail.com

More than Meets the Eye: A Closer Look at Encodings in Visualization

Robert Kosara
Tableau Research

***Abstract*—Encoding data visually is at the heart of visualization. We usually assume that encodings are read as specified, i.e., if a bar chart is drawn by the length of the bars based on the data, that is also how we read them. We question this assumption and demonstrate that observed encodings often differ from the ones used to specify the visualization. The value of a chart also often comes from higher-level derived encodings, and which encodings end up getting used also depends on the user's task.**

ENCODING DATA into visual properties is a key part of data visualization. Bars encode values by length, scatterplots and line charts by position, pie charts by angle, etc.

The implicit assumption here is that the way we specify and construct charts and visualizations is also how we read them. But charts are no longer used to store data, letting us read individual values off of them is not what makes them useful. Rather, we tend to look for patterns, shapes, and trends. A line chart may encode the number of vaccinated people in the position of its vertical distance from the baseline, but what we are looking at is whether the slopes of the lines between those points increase or decrease over time. A scatterplot may encode two data dimensions in the locations of the dots, but is useful because it allows us to see correlations and spot outliers. A histogram encoding counts in the lengths of its bars is read mostly as a shape.

The visualization literature is full of studies investigating the effectiveness of different charts, but the perceptual mechanisms are often just assumed. Do we read bar charts by length or area? Do we read bubble charts by area or diameter? How *do* we read pie charts? Perhaps bar charts are more complicated than we think, which would explain the impact of aspect ratio on visual judments [4], for example.

In some cases, however, the specified encodings might not even be among the ones that we read. Pie charts in particular are specified by angle in software, since they are drawn as circle sectors or wedges. It seems reasonable to assume that angle would be the way we then also read them, but a series of studies has found that arc length or area are more likely to be the visual cue we read [9].

 Published by the IEEE Computer Society

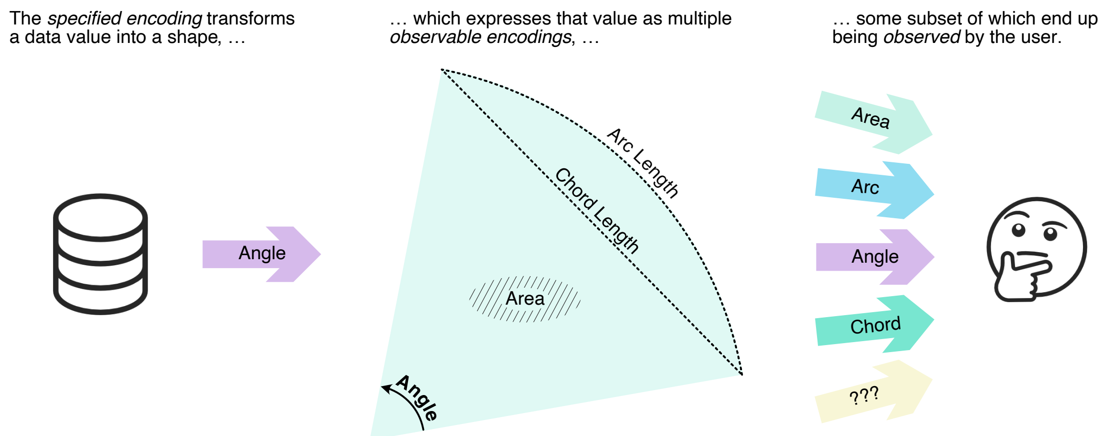

Figure 1. Pie charts are specified by angle, but may also be read by area, arc length, or even chord. Shape recognition is also likely for specific angles like 90°/25% and 180°/50%.

In this paper, we argue for a clear distinction between different kinds of encodings: the specified (input) encodings that we use to draw visualizations; the observable basic visual features a user could be reading; those that are actually observed; as well as high-level derived information that is not specified or observed directly, but can be extracted from the chart. We suggest that this approach provides not only a clearer understanding of how visualization works and is used, but also presents interesting opportunities for research that investigates the different encodings of charts and how they relate to each other.

Encodings: Specified, Observed, Derived

The idea of encodings goes back at least to Bertin's *retinal variables* [1], which express numbers in visual properties such as position, size, color, etc. They were famously used in subsequent work by Mackinlay [7] to rank encodings and build an automated visualization system, as well as in an attempt at structuring the visualization space around encodings [3].

Encodings are usually assumed to be read exactly as specified, but we argue that they need to be considered separately. What is specified might not be what is read, only or at all. We also believe that what viewers look for depends on the task, such as in a line chart being read as a trend.

Specified encodings are those mappings of data to visual properties that are used to create the chart or visualization. These are what are currently considered encodings. They include the length of a bar in a bar chart, the angle of a pie slice in a pie chart, the horizontal and vertical position of a point on a line chart, etc.

Observable encodings are those base-level visual properties of the visualization that a viewer can perceive. They may or may not overlap the set of specified encodings. They include properties such as area and aspect ratio in addition to length in a bar chart, area and arc length in a pie chart, line slope and length in a line chart, etc.

Observed encodings are the observable encodings that the user actually ends up taking from the visualization. Which ones are observed can depend on the task, and in most cases, there is no conclusive evidence for one particular encoding as the observed one out of the set of observable ones. For this discussion, we therefore treat all output encodings as observable rather than observed, and leave the distinction to further research.

Derived encodings are higher-level information derived from the base encodings observed from the chart. This set is separate from specified and observed encodings, and depends more heavily on the viewer's task. They include the overall shape (envelope) of the bars in a histogram, the trend of a line in line chart spanning multiple points, the correlation between the axes in a scatterplot, etc.

The distinction between specified and observable encodings might appear like a technicality,

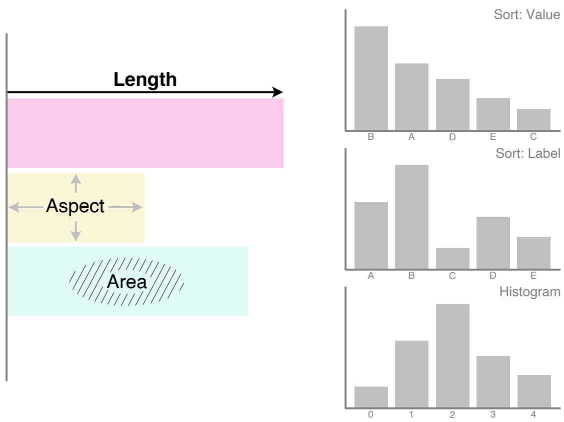

Figure 2. Bar charts encode values as their length, but also their area, aspect ratio, and overall shape. Sorting is often used for specific use cases.

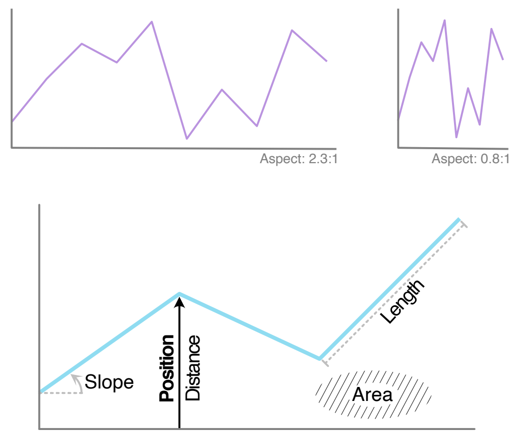

Figure 3. Line charts are specified by the location of the points connected by lines, but are read as slope and length, as well as area. Aspect ratio of the chart is also generally considered important.

since in many cases they behave the same: a bar's area increases linearly with the value it represents, just like its length; a pie slice's area and arc length increase linearly with its angle; etc. This masks the distinction and makes it difficult to isolate what is even being observed. In addition to understanding how our perceptual system works in visualization, knowing the precise mechanism when reading even simple charts is important for cases where some of those visual cues might be missing (such as the central angle in a donut chart) or not be fully consistent with each other (such as in many embellished charts used in information graphics).

Observed and derived encodings are intended to cover the gamut of potential encodings that are available to the viewer in decoding a chart; which are used in any one instance may differ due to the viewer's mental model (as discussed in the case of the treemap below) and task. Task, in fact, appears to be an important, and often overlooked, factor here. If the user is really interested in how much larger the sales figures encoded in a bar chart are between consecutive quarters, for example, length will be the likely encoding used. But the same chart might serve to judge whether sales have been trending up or down over the last multiple years, in which case individual lengths are secondary to the enveloping shape of the entire chart.

Bar Charts, Histograms

Bar charts are seemingly simple, in that they clearly encode data in their length. Even rectangular bars have areas and aspect ratios that change with the represented number, however (Figure 2). How do we know that they are not read by the viewer? Evidence from studying information graphics shows that pictorial bar charts can lead to measurable distortions when the area changes in a way that is incongruent with length [10]. Recent work has also shown that the aspect ratio and shape of bar charts can influence people's ability to read them [4].

Additionally, histograms are usually read as shapes more than individual bars, where their enveloping shape, a derived encoding, is the key. This way of reading is also common in bar charts that are sorted by value where the user then looks for breaks in the overall shape.

Specified: length
Observable: length, area, aspect ratio
Derived: enveloping shape

Line Charts

Line charts encode position in the points—why do we connect them with lines? In fact, most line charts do not seem to draw the points at all, but only the connecting lines. While it is technically possible to retrieve specific point values from line charts, their main use is in judging the rate of change between points, be they adjacent or more distant across the chart.

While not specified, lines in line charts have lengths and slopes (Figure 3). Both allow us to judge the difference in vertical position between adjacent data points, which in the case of regular spacing is a direct indication of the amount of change. When points are not regularly spaced, the slope still shows the rate of change, though in that case the length of the line is also a function of the distance between the points on the horizontal axis.

In addition to single lines, line charts show the behavior of a value over a larger number of samples (typically along a time axis), thus showing the change in slope and direction (and thus, an approximation of the second derivative of the value) over a longer period of time. "Eyeballing" a line chart also allows the user to smooth and aggregate a line chart across many data points to get a rough estimate of the long-term rate of change of the value.

The area under the chart is ignored as an encoding in the case of a line chart, but it appears to be potentially meaningful even when not filled in. The differences between line and area charts might be subtle, especially when the chart appears small or is seen as one of many on a complex dashboard or in a sparkline-style configuration.

Other complications come from line charts containing multiple lines that form areas, present distances that are often of interest (but are known to be difficult to judge accurately), and cross.

The existing view of encodings also does not take into account the role of aspect ratio. If a line were read as points, the difference between charts of different width-to-height ratios would be of little importance. *Banking to 45°* and the role of aspect ratio in lying with visualization suggest otherwise, however.

Specified: position
Observable: position/distance from baseline, slope, area under the line, length of lines between points, line crossings
Derived: line interpolations between non-adjacent points, enveloping shape, line distances

Pie Charts

The common assumption for pie charts is that we read them by angle. They are clearly specified and constructed by angle; were they to be specified by areas or arc lengths, those would have to be converted into angles to be drawn. But they also express the value in the area and arc length of the slice, as well as the length of chord (Figure 1).

We showed in several studies that angle is, in fact, the least likely visual cue being used to read pie charts [9]. While it is understandable that we assume that the specified encoding is also the one that is being read, such an assumption needs to be tested. Assumptions lead to further assumptions: if we read a pie chart by angle, then clearly a donut chart, which is missing the center where we can best see that angle, must be even worse than a pie chart! That would seem reasonable if there were evidence for angle as the mechanism, but that does not appear to exist. Our studies also showed donut charts to be indistinguishable from pie charts in reading accuracy, while larger slices (which increase area and arc length, but not angle) led to over-estimation of the represented value.

Another reason making the angle assumption attractive is that certain values are very recognizable in a pie chart, such as 25% (right angle) and 50% (straight line). We believe that these are not primarily read by angle either, but rather recognized as shapes.

Specified: angle
Observable: angle, area, length of arc, length of chord
Derived: shape

Parallel Coordinates

Similar to line charts, parallel coordinates specify position of the points on each axis that are connected by lines. Their utility does not come from this simple encoding however, but from the patterns that are easy to spot from them: how parallel are the lines between a pair of axes? How busy does the area between two axes appear (i.e., how many line crossings are there), etc.

Similar to scatterplots, parallel coordinates allow users to judge the level of correlation between axes based on these observed and derived encodings [5]. The specified encoding, position, is all but irrelevant to how parallel coordinates are used and read.

Specified: position
Observable: position, slope, lengths of lines connecting points, line crossings

Derived: parallelism between the lines, correlation between axes, clutter between axes

Treemaps

Treemaps translate a hierarchical data structure into a two-dimensional layout. They specify area as the way to represent the numerical value, but location is more complicated. While the layout is algorithmically derived from the hierarchy, it is not specified in the same sense as the length of a bar in a bar chart. The choice of algorithm (e.g., the squarified treemap [2]) is just as important in this case as the order of the nodes within each parent node (which are often sorted by size).

How do people read the resulting visualizations? The single-layer treemap that is commonly used in business intelligence today provides a part-to-whole comparison based on area, as well as more or less arbitrary spatial groupings. When the data is actually hierarchical, however, the spatial layout communicates the structure of the tree. In previous work, we found that how people interpret it can differ depending on the framing of the question or the user's mental model, whether they use containment or levels [11].
Specified: area, hierarchy
Observable: area, location, containment
Derived: spatial grouping, hierarchy

Untangling Encodings

In order for studies to lead to conclusions that generalize, we need to know what the user has actually observed. If that is equated with what has been specified as the input encoding without further analysis, the wrong conclusions might be drawn. This is akin to Munzner's threats to validity of evaluations: an evaluation can be meaningless or even misleading if it uses the wrong criteria [8].

We see a rich opportunity for research here that investigates the observed and derived encodings of common visualization and chart types – in particular those that are fully congruent with each other, like the area and length of a bar in a bar chart. Many might turn out to be complex combinations of multiple encodings (as appears likely in the bar and pie charts) that depend on the task just as much as on the visualization technique itself. We believe that this will lead to a better understanding of how to construct visualizations that are the best suited for a particular user in a given situation.

Different observable encodings can be difficult to untangle because they often have linear relationships with each other (such as area and length in bars, etc.). One possible way to separate them is to introduce distortions, such as by using three-dimensional versions of the charts [6]. The resulting distortion can be modeled, and is uniform when orthogonal projection is used. More research approaches will need to be developed to get deeper insights into this question.

Conclusion

Encodings are more complex than is often assumed, and we believe that drawing a distinction between what is specified and what can actually be observed, with an extra dependency on task, will help us gain a better understanding of charts and visualization techniques.

Even—or perhaps especially—simple chart types like bar and line charts, pie charts, etc. deserve another look to better understand how they really work, and to develop a more rigorous scientific approach to understanding and developing visualization techniques and tools.

REFERENCES

1. J. Bertin. *Semiology of Graphics*. University of Wisconsin Press, 1983.
2. M. Bruls, K. Huizing, and J. J. van Wijk. Squarified Treemaps. In *Data Visualization (Proceedings VisSym)*, pages 33–42. Springer, 2000.
3. S. K. Card and J. Mackinlay. The Structure of the Information Visualization Design Space. In *Proceedings Information Visualization*, pages 92–99. IEEE CS Press, 1997.
4. C. R. Ceja, C. M. McColeman, C. Xiong, and S. L. Franconeri. Truth or Square: Aspect Ratio Biases Recall of Position Encodings. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1054–1062, 2021.
5. L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking Visualizations of Correlation Using Weber's Law. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1943–1952, 2014.
6. R. Kosara. Evidence for Area as the Primary Visual Cue in Pie Charts. In *IEEE VIS Short Paper Proceedings*, 2019.

7. J. D. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *Transactions on Graphics*, 5(2):110–141, 1986.
8. T. Munzner. A Nested Model for Visualization Design and Validation. *Transactions on Visualization and Computer Graphics*, 15(6):921–928, 2009.
9. D. Skau and R. Kosara. Arcs, Angles, or Areas: Individual Data Encodings in Pie and Donut Charts. *Computer Graphics Forum*, 35(3):121–130, 2016.
10. D. Skau and R. Kosara. Readability and Precision in Pictorial Bar Charts. *Short Paper Proceedings of the Eurographics/IEEE VGTC Symposium on Visualization (EuroVis)*, 2017.
11. C. Ziemkiewicz and R. Kosara. The Shaping of Information by Visual Metaphors. *Transactions on Visualization and Computer Graphics*, 14(6):1269–1276, 2008.

Robert Kosara is lead research staff at Tableau Research and a former professor of Computer Science at UNC Charlotte. His research interests revolve around the uses of visualization for communication: how do we read charts, how do we use unusual chart types to capture attention and help a message stick in the viewer's memory, how do presentation and storytelling with data work in different contexts, etc. Contact him at rkosara@me.com.