Editor:
Theresa-Marie Rhyne

In Color Perception, Size Matters

Maureen Stone
StoneSoup Consulting

Most studies on the effective use of color for visualization evaluate different colorings on objects of the same size. This is good practice for experimentation because it minimizes the number of variables being studied. But color appearance changes dramatically with the size of the object being viewed. Anyone who has picked a paint color from a sample chip and been surprised as to how much more colorful it appeared on the wall has experienced this effect. My experience designing colors for digital visualization systems suggests that the interaction between size and color appearance is a critical design factor. Whereas artists and visual designers understand this, recommendations for using color in computer-generated visualization rarely address the issue, or even recognize it. I write this to bring the topic to the community's attention, provide practical guidelines, and possibly inspire research in this area.

Consider the five colors in Figure 1, which are different hues of the same luminance (perceived brightness) and similar colorfulness, as is typically recommended for categorical labeling.[1,2] Shown as half-inch squares, they are distinct and visually balanced. You could easily use these colors to color bars in a bar chart or the regions of a choropleth map, such as those used for census data. The same colors applied to a line graph, however, are less distinct, especially between blue-green and yellow-green. In the scatter plot, the colors are even harder to distinguish.

The problem becomes even more severe for lighter, more pastel colors. In Figure 2, the light shading colors in the table are easily distinguished. If anything, the strongest colors are darker and more colorful than are ideal for display behind text. Consider, however, the same colors displayed as little squares in the adjoining color picker. Which correspond to the colors in the table? The arrow marks the column containing the strongest colors, with the two lighter steps of the same color to the right. The strongest colors are just visibly different, the lighter values nearly indistinguishable. Only the context makes guessing the color of the lighter steps possible.

Physical and Psychophysical Factors

So what is going on in these examples? There are both physical and psychophysical effects to consider. Small features antialiased in rendering will have the adjacent background blended with the foreground color. As a result, the rendering physically changes the color to one that is a blend of the foreground and background. Even without antialiasing, light between adjacent display pixels can mix. So, a single colored pixel surrounded by a background color is effectively emitting light of a different color than one embedded in a field of pixels of the same color.

A similar blending process also occurs in the human visual system, which is spatially sampling and integrating as our eyes scan. Vision science calls this effect *spreading*. This is typically illustrated as in Figure 3, in which the colors of the distinctly cyan and yellow stripes start blending as the stripes become smaller (higher frequency). In the limit, the distinction between adjacent colors vanishes, as it does in pointillist paintings or halftone printing.

Color science offers little help in computing the effect of size on color perception. The standard CIE (International Commission on Illumination) metrics for color matching and measurement are based on observations of 2-degree samples (the 1931 standard observer). Two degrees is fairly large compared to many digital-imaging features, roughly the size of your thumb joint held at arm's length. (The samples in Figure 1 will be 2 degrees if viewed at 15 inches.) Two degrees is the area viewed by the fovea, the part of the retina that has the highest density of cones. The CIE established a 10-degree observer (about the size of your fist at arm's length) in 1964 and recommends using it for samples larger than 4 degrees. In practice, however,

0272-1716/12/$31.00 © 2012 IEEE

the 1931 observer is widely used for samples of all sizes, including in the formulation of the color difference metrics CIELAB and CIELUV.

Researchers have modeled spreading, but only for objects as small as a pixel or halftone dot. Consider the problem of automatically comparing two images, one of which has undergone processing such as dithering or halftoning. Compared pixel by pixel, the images would be very different, although visually, they might be quite similar. The S-CIELAB color difference model uses filtering to mimic the spatial integration that occurs with pixel-sized samples.[3] Images are compared by converting their pixel values from CIE tristimulus values (XYZ) to a representation closer to that of the cone responses and applying a low-pass filter to blend adjacent values. This creates similar pixel values for visually similar images.

Implications for Visualization

Figure 4 shows a set of categorical colors that differ more distinctly at small sizes than those in Figure 1. Overall, they are darker and more colorful, and their luminance varies as indicated in the caption. In the line graph, the colored lines are now quite distinct. In the scatter plot, the luminance variation gives the strongest cues. The dark blue dots stand out strongly, as do the light cyan ones. However, it is more difficult to distinguish the orange dots from the pink ones, or even the green from the blue.

These scatter plot examples show that color alone cannot reliably label items as small as the individual dots, even for as few as five categories. However, the coloring does indicate clusters and trends, even for much higher numbers of colors. I would argue that most colorings applied to scatter plots and images do not robustly show unique values on features the size of a pixel, but instead show clusters or regions of similar or contrasting values. These larger aggregations of color will be usefully visible. For example, discussions of effective pixel-based techniques emphasize the need to organize the data to create clusters.[4]

Designing colors for different-sized objects is a balancing act; colors must be strong enough to work for small objects but not overwhelming for larger ones. If the size range is extreme, adjusting the coloring to the size might be necessary. Research in models to do this automatically would be a useful contribution to the visualization literature.

When designing colors for very large regions, the challenge is to keep the color from being too dark and too colorful, especially when it is used behind other visual elements, such as text in a table (see Figure 2), or when it highlights a region in a graph. On a white background, well-designed background colors have L* values in the range of 90 to 96 (lower for blue and purple, higher for

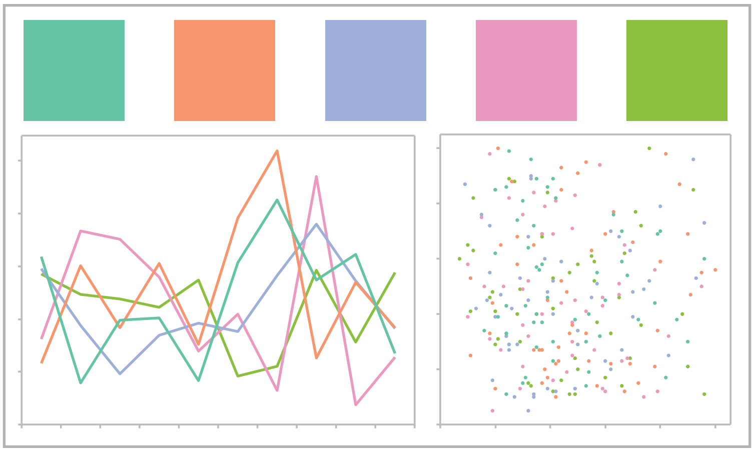
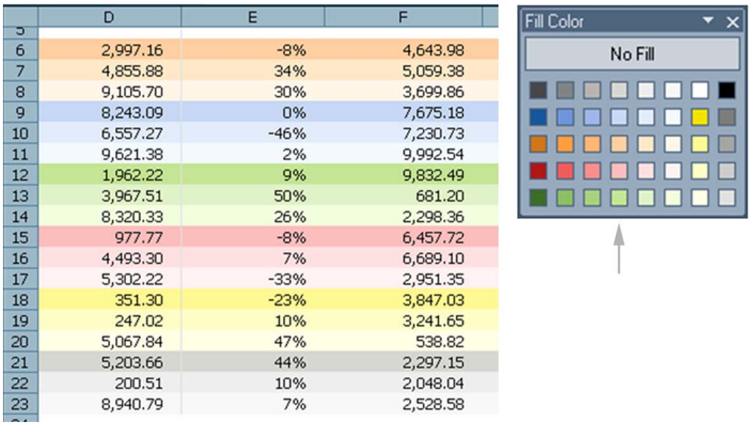


Figure 1. A set of colors with different hues but equal luminance (L* = 72). In the squares, these colors are distinct and visually balanced. However, in a line graph or scatter plot, they are harder to distinguish. The colors in this figure are adapted from one of the ColorBrewer qualitative color sets (www.colorbrewer.org).



	D	E	F
6	2,997.16	-8%	4,643.98
7	4,855.88	34%	5,059.38
8	9,105.70	30%	3,699.86
9	8,243.09	0%	7,675.18
10	6,557.27	-46%	7,230.73
11	9,621.38	2%	9,992.54
12	1,962.22	9%	9,832.49
13	3,967.51	50%	681.20
14	8,320.33	26%	2,298.36
15	977.77	-8%	6,457.72
16	4,493.30	7%	6,689.10
17	5,302.22	-33%	2,951.35
18	351.30	-23%	3,847.03
19	247.02	10%	3,241.65
20	5,067.84	47%	538.82
21	5,203.66	44%	2,297.15
22	200.51	10%	2,048.04
23	8,940.79	7%	2,528.58

Figure 2. The shading colors are easy to distinguish in the table but difficult to distinguish in the small color picker. The arrow indicates the darkest colors shown in the table; lighter shades are to the right.

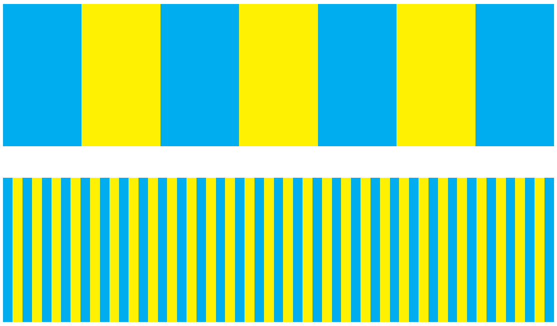


Figure 3. An example of spreading, demonstrating that the human eye blends small areas of color that are close together. The alternating cyan and yellow colors are the same in both bars.

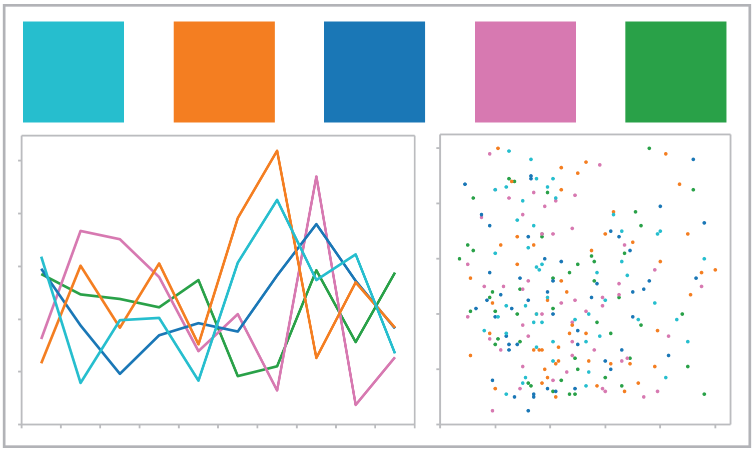


Figure 4. These colors vary in luminance, which makes them more distinguishable when rendered small. From left to right, L* = 70, 68, 47, 65, and 58.

Table 1. ΔL* guidelines for text readability.*

ΔL*	Description
10	Easily visible but difficult to distinguish similar shapes
20	Legible but not comfortably readable
30	Easily readable, similar to the ISO recommendation of 3:1
60	Robustly readable, similar to the ISO recommendation of 10:1
100	Maximum contrast, black and white

ΔL *is the absolute value of the difference between the* L* *values of the foreground color and the background colors*

yellow and green). On an sRGB display, a pixel value as light as RGB = (240, 240, 240), which has an L* value of 95, is a usefully visible color for such a background and seems fairly robust across most modern LCD displays (although less so for projectors). Small variations around this value can produce a set of visibly different colors suitable for large areas. As Figure 2 also shows, creating an effective UI for these ultralight colors can be challenging. I speculate that one reason background colors are consistently too dark and vivid in applications such as Excel is that they are designed in the context of their UI rather than their application.

Some New Approaches

Rather than basing our research in color for visualization on purely perceptual characteristics, we need to design coloring in the broader context of the visualization system. Rather than providing a unique label for categories or quantity, coloring is simply one part of a dynamic exploration, in which tooltips and the ability to filter and recolor also assist users in exploring their data.

In designing colors for data, increasing colorfulness and introducing luminance variation makes colors easier to distinguish and helps make the colors more visible to people with less than perfect color vision. People with the more common forms of color vision deficiency (CVD) have normal luminance perception, often enhanced by their reduced color perception. Many can see colors that are strong and colorful but have difficulty with pastel or grayish ones. (For more information on designing for CVD, see *Guidelines for Using Color in Voting Systems.*[5])

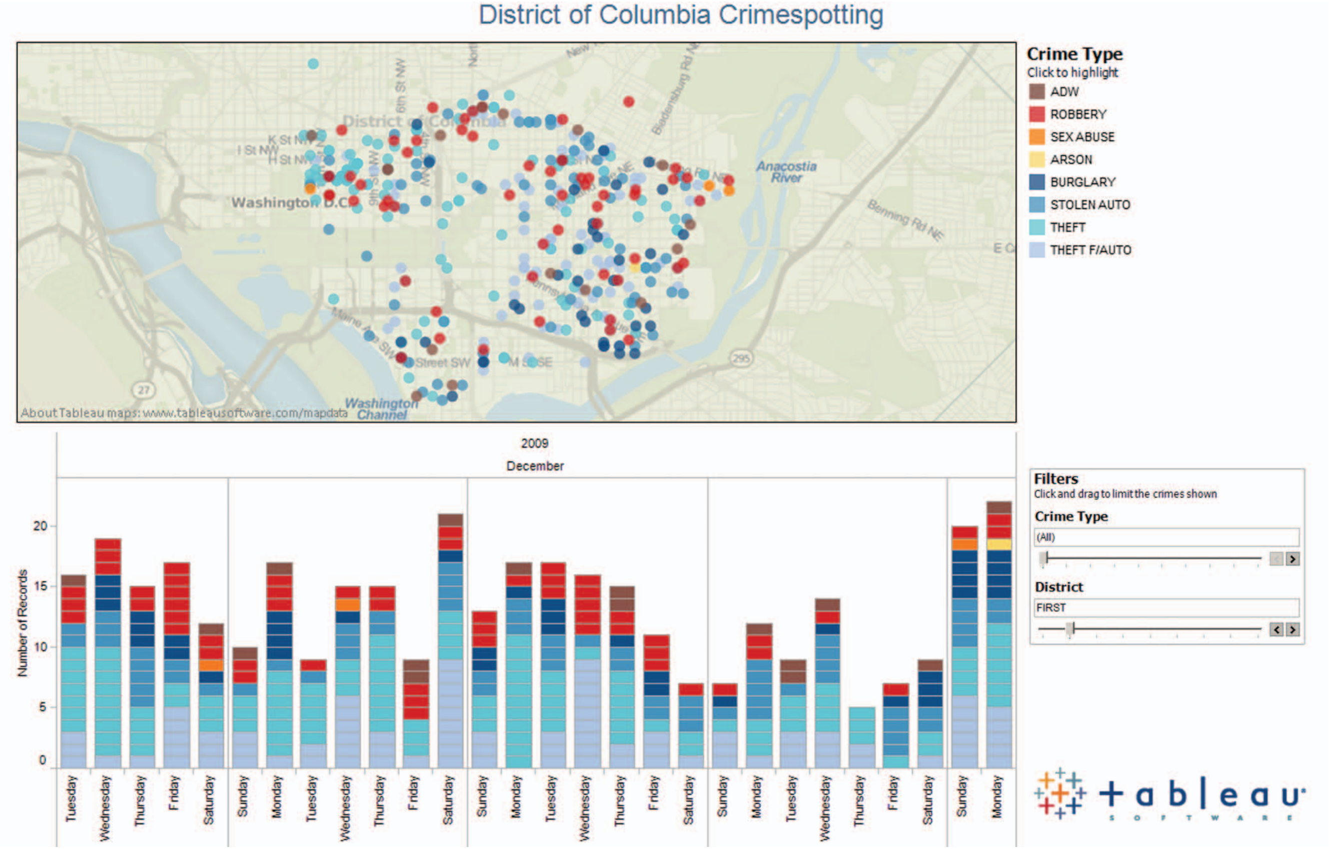
Varying luminance and colorfulness also helps to make full use of the available color gamut. It could be argued that the common practice of using the corners of the RGB color cube is effective because it maximizes luminance variation and optimizes the use of the gamut. A better goal, however, is to achieve a balance, in which the colors appear to have similar visual weight while including some luminance variation, as in Figure 4.

The most critical limits on color are constraints on legibility, which is controlled almost entirely by size and luminance contrast. Smaller characters need higher contrast to be readable. ISO legibility standards are stated as luminance ratios (for example, 3:1 and 10:1). Table 1 shows metrics based on ΔL*, computed as the absolute value of the difference between the L* values of the foreground color and the background color. An advantage of using ΔL* instead of a luminance ratio is that you can use Adobe Photoshop to select colors with specific L* values. The guidelines are for a typical text label or table entry (for example, 9-point Ariel).

Tableau Example

Tableau Software designs business analytics systems that provide an innovative visualization front end to enterprise-sized databases (www.tableausoftware.com). Visualizations include text tables, scatter plots, line graphs, bar graphs, and data maps. The graphs in Figures 1 and 4 were created using the Tableau Desktop product; Figure 5 shows some other examples. Working with Tableau developers, I designed the colors used in these applications and contributed to the design of their user interfaces for managing and applying color.

For categorical coloring, the goal was to create the maximum number of colors that were legible, identifiable, and easily named. We found that a set of 10 basic colors for labeling (categorical colors) and a second set of 20 colors consisting of the 10 basic colors plus a lighter set of those colors met this goal. If the user has more than 20 categories, the system gives a warning and reuses the colors. Users can also select from other palettes and, if necessary, use the system color picker to select



Crime Type: (All)

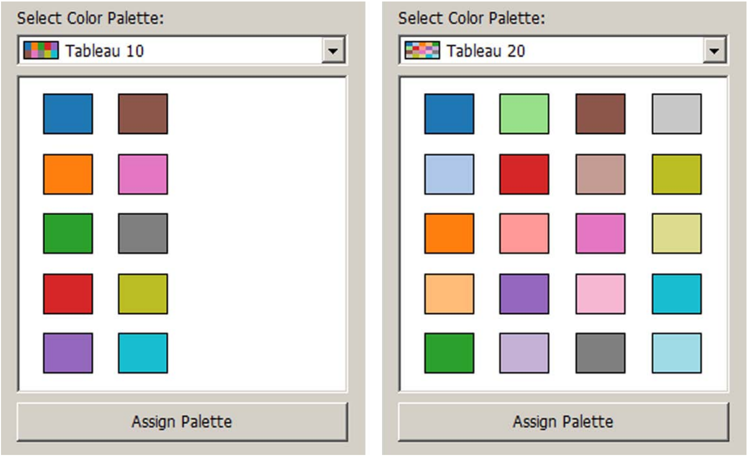
District: FIRST

Figure 5. An example showing several visualizations created using Tableau Software's Desktop product. Tableau creates business analytics systems that provide an innovative visualization front end to enterprise-sized databases.

their own colors.

Users can also use shapes to indicate categories; Tableau provides tools that allow quick selection and filtering of categories. Tooltips let users easily query individual items for their specific data values. Users can also select items by name in the color legend, to display only the selected items in color, reducing all others to shades of gray.

Users employ a palette-based interface to apply colors, which are by default assigned in the order shown in Figure 6, top to bottom, left to right. For the Tableau 20 palette (see Figure 6b), the system intermixes light and dark colors to balance the number of light and dark colors used. The colors were designed visually in Photoshop but tuned by analysis in CIELAB space using custom tools written in Python. The colors in Figure 4 are from the Tableau 10 palette (see Figure 6a).


Select Color Palette: Tableau 10

Select Color Palette: Tableau 20

Figure 6. The (a) Tableau 10 and (b) Tableau 20 colors, shown in the palette-based UI. Users can assign the entire palette or select individual colors and map them to items in a list (not shown).

Figure 7 shows the Tableau 20 colors plotted in the CIELAB space, L*a*b*. While a* and b* are color difference values (roughly red-green and blue-yellow), the a*b* plot (see Figure 7a) can be more usefully viewed as a hue circle, with the achromatic axis in the center, the hue defined by the angle, and the chroma (saturation) increasing radially. For comparison, Figure 7a also shows a representation of the gamut boundary, which interpolates the primary and secondary colors (corners of the RGB color cube). As you can see, the light Tableau colors have a lower chroma than the dark ones, a constraint of the color gamut. Brown is a darker, lower-chroma version of red-orange. In the L* stacked diagram (see Figure 7b), the white background has an L* value of 100. The dark colors are all easily readable (L* < 80), but some of the light

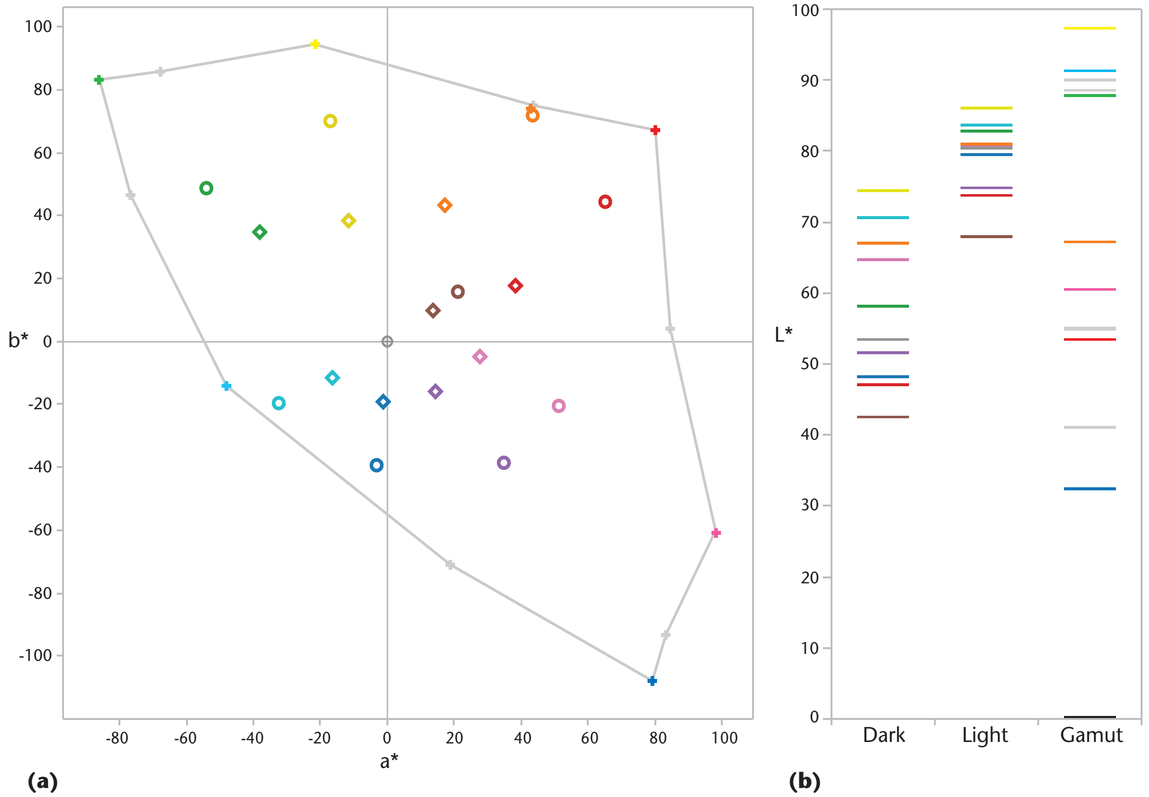


Figure 7. The Tableau 20 colors plotted in CIELAB space, L*a*b*. (a) Colors projected onto the a*b* plane. The dark colors (Tableau 10) are circles, the light ones (added for Tableau 20) are diamonds, and each is colored by its corresponding hue. The corners of the RGB color cube are plotted as small crosses, and connecting lines have been drawn to represent the gamut boundary. (b) The L* value for each color in Figure 7a plotted as a horizontal line. (Dark and light versions of each Tableau color are the same dark color in both plots, for legibility.)

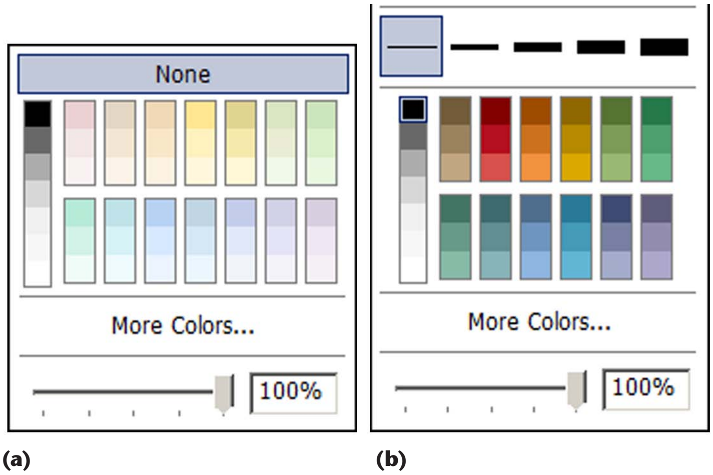


Figure 8. The Tableau formatting colors for (a) shading and (b) annotation. The slider controls transparency, which can be used to lighten the colors.

ones fall into the "legible but not comfortable to read range" (80 < L* < 90) and are therefore less suitable for text. The infamous HSV (hue, saturation, and value) rainbow color map would traverse the gamut boundary shown and would range in L* value from 97 (yellow) to 32 (blue).

A different set of colors called "formatting colors" were designed for shading (very light, for large areas; see Figure 8a) and annotation (dark, for text and lines; see Figure 8b). These colors were chosen to be aesthetically different than the data colors and optimized for their specific application. Again, they were visually designed but analyzed and tuned algorithmically. These colors are arranged in groups of three so that for the pale shading colors, the darkest colors can label the lighter ones.

The darkest of the light colors range in L* value between 85 and 90, the lightest step is 96 to 98, and the middle step is roughly halfway between. For the dark colors, the darkest step is around L* = 45 and the lightest is around L* = 70. The exception is red, which is much darker. The slider controls transparency, which can be used to lighten the colors. Clicking on "More Colors ..." will drop into the Windows color picker, but most users are satisfied with the set provided.

Visualization systems map color to a property of the data, such as its category or value. This article has focused on categorical colors, but the issue of size is also significant for color scales used for numeric values. This viewpoint has two primary messages. First, size matters, and an opportunity exists for more research in effective color for visualization around this factor. Second, color in visualization systems is designed in a broader context than color perception or color design. The goal should be to keep the user focused on the data, with color simply one of many tools used toward problem solving and discovery. We are only beginning to learn how to optimize for this larger goal.

References

1. C.G. Healey, "Choosing Effective Colours for Data Visualization," *Proc. 7th Conf. Visualization* (VIS 96), IEEE CS Press, 1996, pp. 263–270; doi:10.1109/VISUAL.1996.568118.
2. R. Ihaka, "Colour for Presentation Graphics," *Proc. 3rd Int'l Workshop Distributed Statistical Computing* (DSC 03), 2003; www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/Ihaka.pdf.
3. X.M. Zhang and B.A. Wandell, "A Spatial Extension of CIELAB for Digital Color-Image Reproduction," *SID Symp. Digest of Tech. Papers*, vol. 27, Soc. for Information Display, 1996, pp. 731–734; http://white.stanford.edu/~brian/scielab/scielab3/scielab3.html.
4. D.A. Keim, "Designing Pixel-Oriented Visualization Techniques: Theory and Applications," *IEEE Trans. Visualization and Computer Graphics*, vol. 6, no. 1, 2000, pp. 59–78.
5. M.C. Stone, S.J. Laskowski, and S.Z. Lowry, *Guidelines for Using Color in Voting Systems*, NIST Interagency Report 7537, US Nat'l Inst. of Standards and Technology, Oct. 2008; www.nist.gov/itl/vote/upload/NISTIR-7537.pdf.

Maureen Stone *works as StoneSoup Consulting and is an adjunct professor at Simon Fraser University's School of Interactive Arts + Technology. She also recently accepted a job at Tableau Software. Contact her at stone@stonesc.com.*

Contact department editor Theresa-Marie Rhyne at theresamarierhyne@gmail.com.