Chapter 20. Redundant Coding

In Chapter 19, we saw that color cannot always convey information as effectively as we might wish. If we have many different items we want to identify, doing so by color may not work. It will be difficult to match the colors in the plot to the colors in the legend (Figure 19-1). And even if we only need to distinguish two or three different items, color may fail if the colored items are very small (Figure 19-11) and/or the colors look similar for people suffering from color-vision deficiency (Figures 19-7 and 19-8). The general solution in all these scenarios is to use color to enhance the visual appearance of the figure without relying entirely on color to convey key information. I refer to this design principle as *redundant coding*, because it prompts us to encode data redundantly, using multiple different aesthetic dimensions.

Designing Legends with Redundant Coding

Scatterplots of several groups of data are frequently designed such that the points representing different groups differ only in their color. As an example, consider Figure 20-1, which shows the sepal width versus the sepal length of three different *Iris* species. (Sepals are the outer leaves of flowers in flowering plants.) The points representing the different species differ in their colors, but otherwise all points look exactly the same. Even though this figure contains only three distinct groups of points, it is difficult to read even for people with normal color vision. The problem arises because the data points for the two species *Iris virginica* and *Iris versicolor* intermingle, and their two respective colors, green and blue, are not particularly distinct from each other.

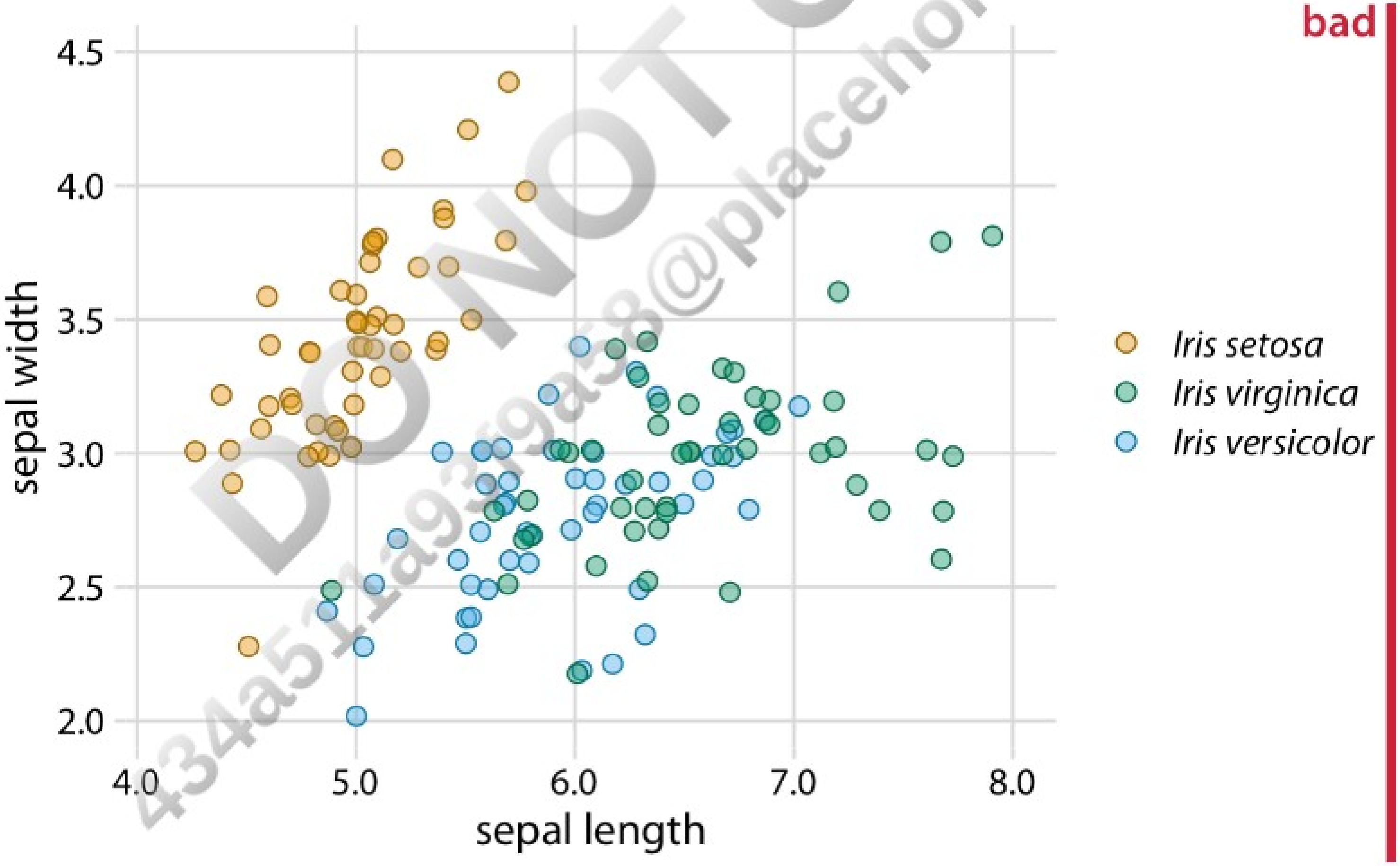


Figure 20-1. Sepal width versus sepal length for three different Iris species (Iris setosa, Iris virginica, and Iris versicolor). Each point represents the measurements for one plant sample. A small amount of jitter has been applied to all point positions to prevent overplotting. The figure is labeled "bad" because the virginica points in green and the versicolor points in blue are difficult to distinguish from each other. Data source: [Fisher 1936].

Surprisingly, the green and blue points look more distinct for people with red–green color-vision deficiency (deuteranomaly or protanomaly) than for people with normal color vision (compare Figure 20-2, top row, to Figure 20-1). On the other hand, for protanomaly) than for people with normal color vision (compare Figure 20-2, top row, to Figure 20-1). On the other hand, for people with blue–yellow deficiency (tritanomaly), the blue and green points look very similar (Figure 20-2, bottom left). And if we print out the figure in grayscale (i.e., we *desaturate* the figure), we cannot distinguish any of the *Iris* species (Figure 20-2, bottom right).

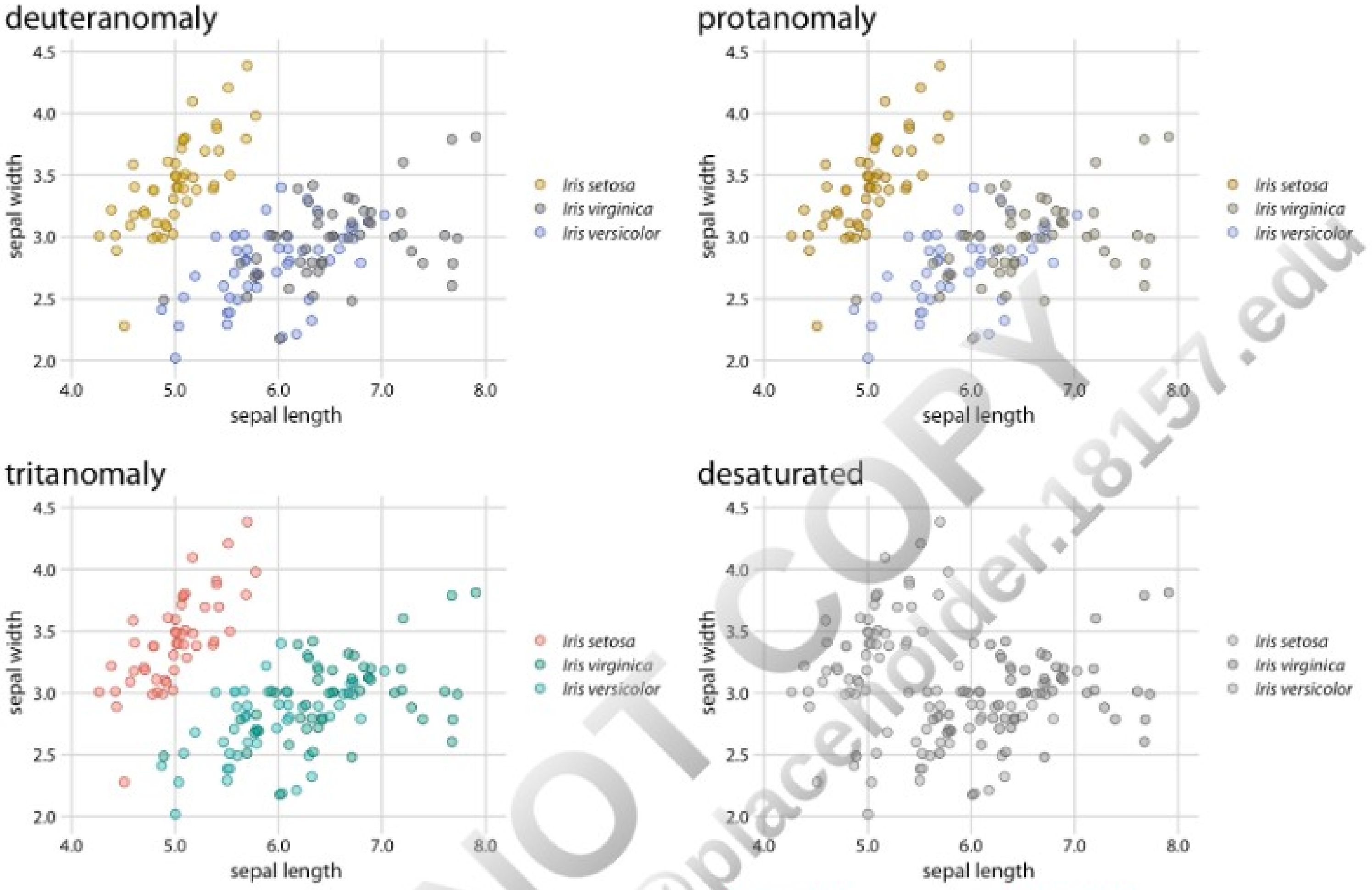
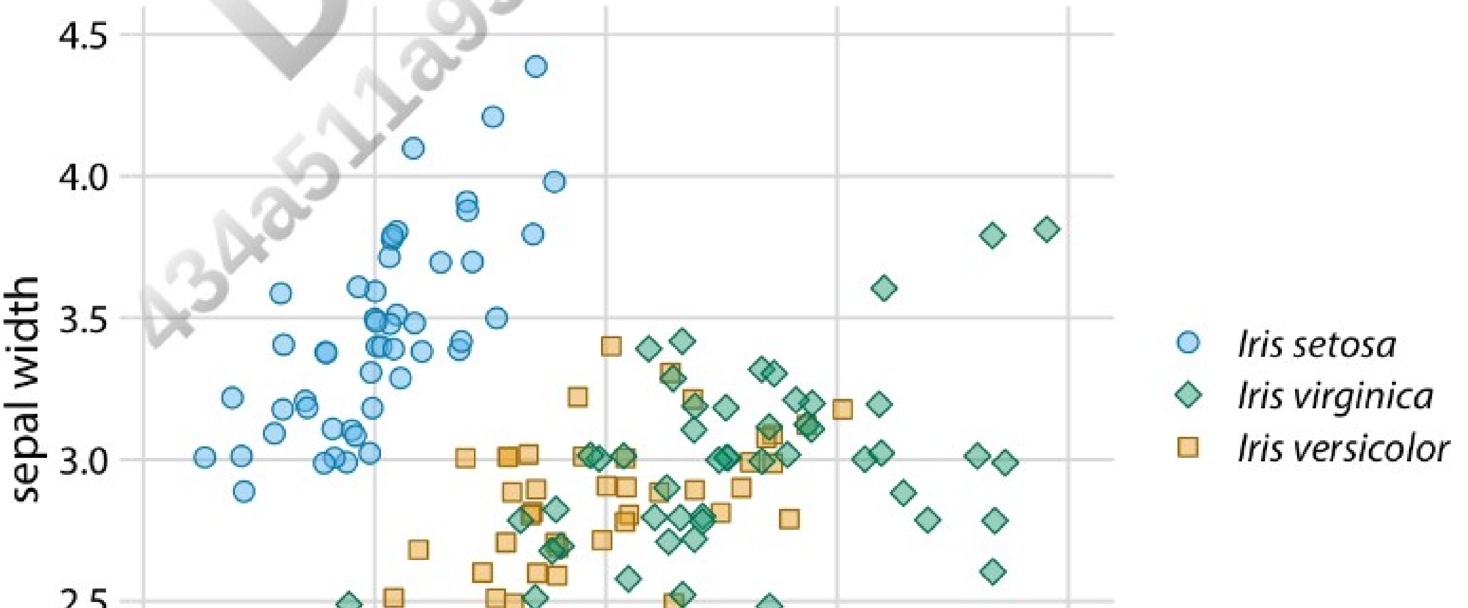


Figure 20-2. Color-vision deficiency simulation of Figure 20-1. Data source: [Fisher 1936].

There are two simple improvements we can make to Figure 20-1 to alleviate these issues. First, we can swap the colors used for *Iris setosa* and *Iris versicolor*, so that the blue is no longer directly next to the green (Figure 20-3). Second, we can use three different symbol shapes, so that the points all look different. With these two changes, both the original version of the figure (Figure 20-3) and the versions under color-vision deficiency and in grayscale (Figure 20-4) become legible.



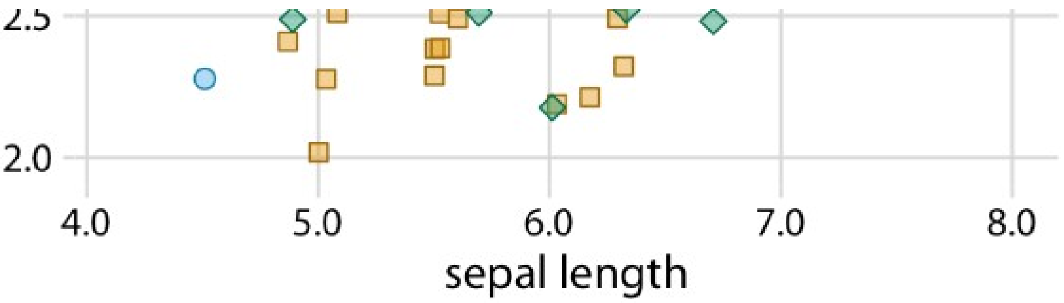


Figure 20-3. Sepal width versus sepal length for three different Iris species. Compared to Figure 20-1, we have swapped the colors for Iris setosa and Iris versicolor and we have given each Iris species its own point shape. Data source: [Fisher 1936].

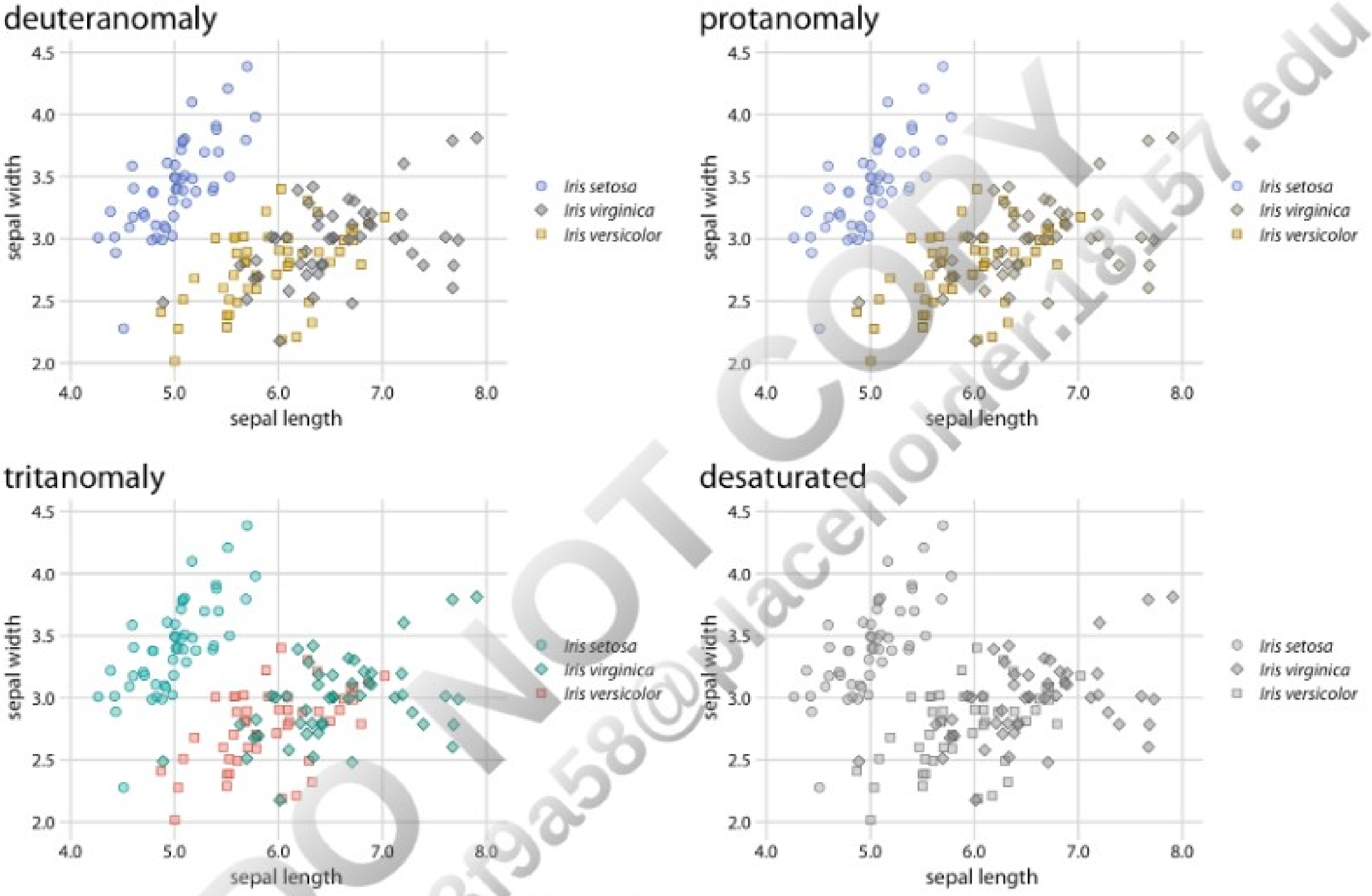


Figure 20-4. Color-vision deficiency simulation of Figure 20-3. Because of the use of different point shapes, even the fully desaturated grayscale version of the figure is legible. Data source: [Fisher 1936].

Changing the point shape is a simple strategy for scatterplots, but it doesn't necessarily work for other types of plots. In line plots, we could change the line type (solid, dashed, dotted, etc.; see also Figure 2-1), but using dashed or dotted lines often yields sub-optimal results. In particular, dashed or dotted lines usually don't look good unless they are perfectly straight or only gently curved, and in either case they create visual noise. Also, it frequently requires significant mental effort to match different types of dash or dot–dash patterns from the plot to the legend. So what do we do with a visualization such as Figure 20-5, which uses lines to show the change in stock price over time for four different major tech companies?

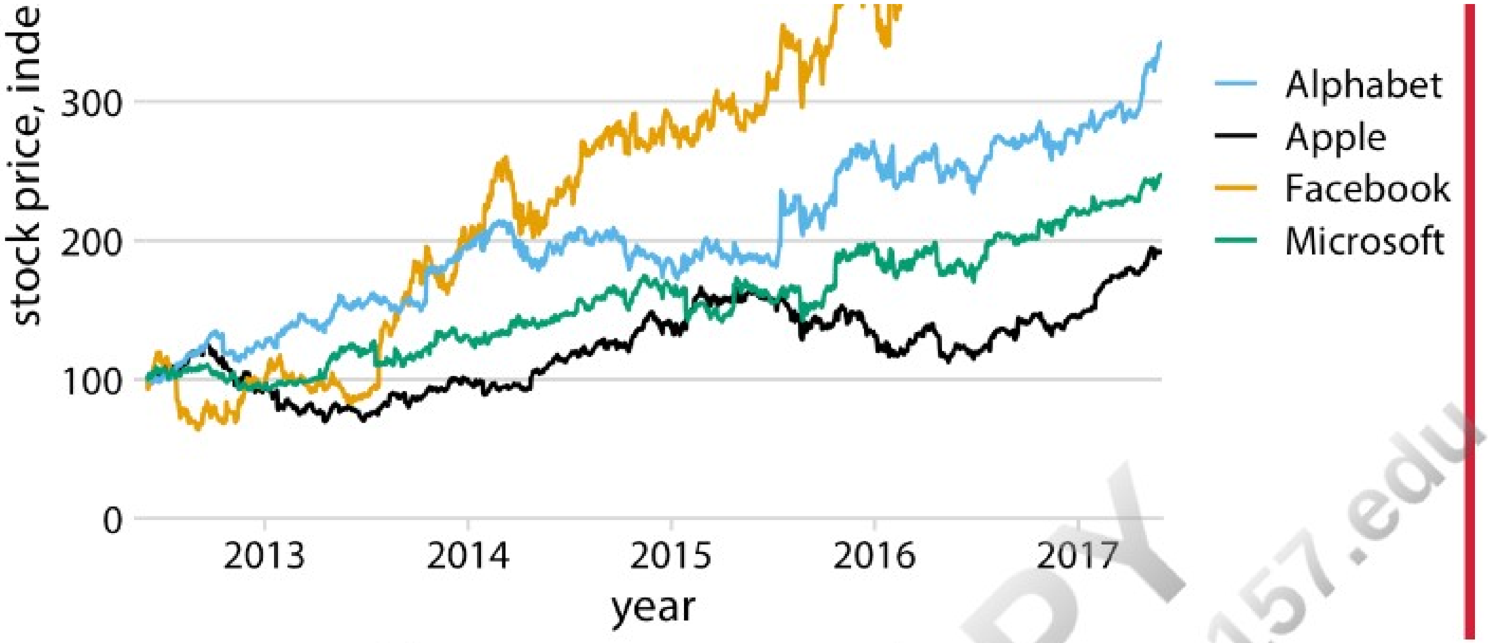


Figure 20-5. Stock price over time for four major tech companies. The stock price for each company has been normalized to equal 100 in June 2012. This figure is labeled as "bad" because it takes considerable mental energy to match the company names in the legend to the data curves. Data source: Yahoo! Finance.

The figure contains four lines representing the stock prices of the four different companies. The lines are color-coded using a colorblind-friendly color scale. Thus, it should be relatively straightforward to associate each line with the corresponding company—yet it is not. The problem here is that the data lines have a visual order. The yellow line, representing Facebook, is perceived as the highest line, and the black line, representing Apple, is perceived as the lowest, with Alphabet and Microsoft in between, in that order. Yet the order of the four companies in the legend is Alphabet, Apple, Facebook, Microsoft (alphabetical order). Thus, the perceived order of the data lines differs from the order of the companies in the legend, and it takes a surprising amount of mental effort to match data lines with company names.

This problem arises commonly with plotting software that autogenerates legends. The plotting software has no concept of the visual order the viewer will perceive. Instead, the software sorts the legend by some other order, most commonly alphabetical. We can fix this problem by manually reordering the entries in the legend so they match the perceived ordering in the data (Figure 20-6). The result is a figure that makes it much easier to match the legend to the data.

This problem arises commonly with plotting software that autogenerates legends. The plotting software has no concept of the visual order the viewer will perceive. Instead, the software sorts the legend by some other order, most commonly alphabetical. We can fix this problem by manually reordering the entries in the legend so they match the perceived ordering in the data (Figure 20-6). The result is a figure that makes it much easier to match the legend to the data.

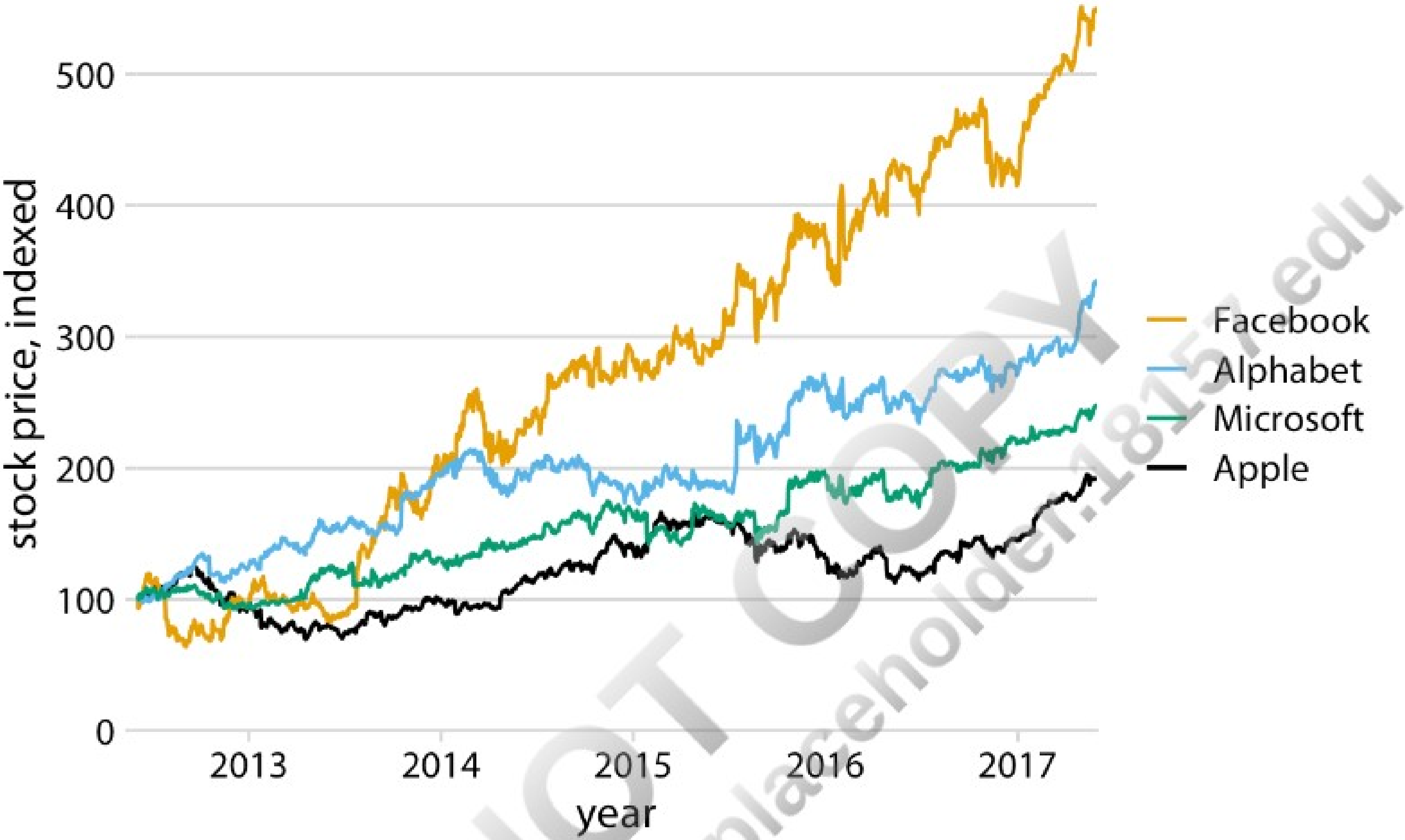
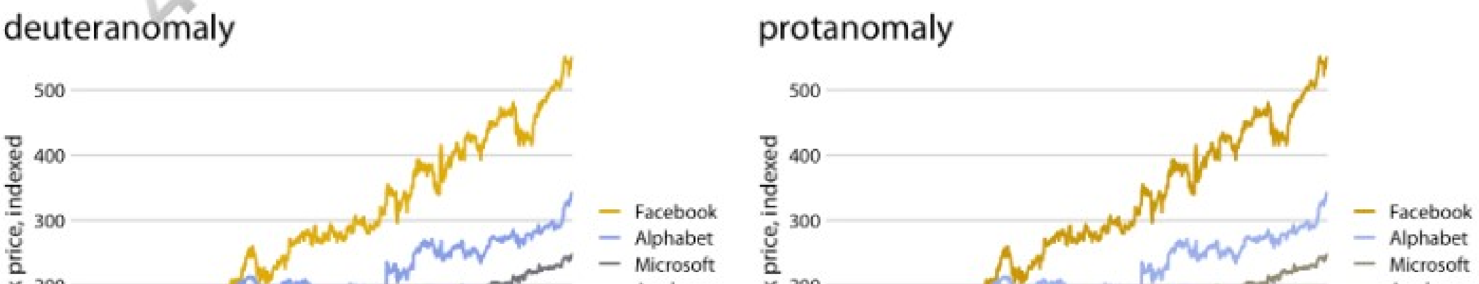


Figure 20-6. Stock price over time for four major tech companies. Compared to Figure 20-5, the entries in the legend have now been ordered such that they match the perceived visual order of the data lines, with Facebook the highest and Apple the lowest. Data source: Yahoo! Finance.

> **NOTE**
>
> If there is a visual ordering in your data, make sure to match it in the legend.

Matching the legend order to the data order is always helpful, but the benefits are particularly obvious under color-vision deficiency simulation (Figure 20-7). For example, it helps in the tritanomaly version of the figure, where the blue and the green become difficult to distinguish (Figure 20-7, bottom left). It also helps in the grayscale version (Figure 20-7, bottom right). Even though the two colors for Facebook and Alphabet have virtually the same gray value, we can see that Microsoft and Apple are represented by darker colors and take the bottom two spots. Therefore, we correctly assume that the highest line corresponds to Facebook and the second-highest line to Alphabet.



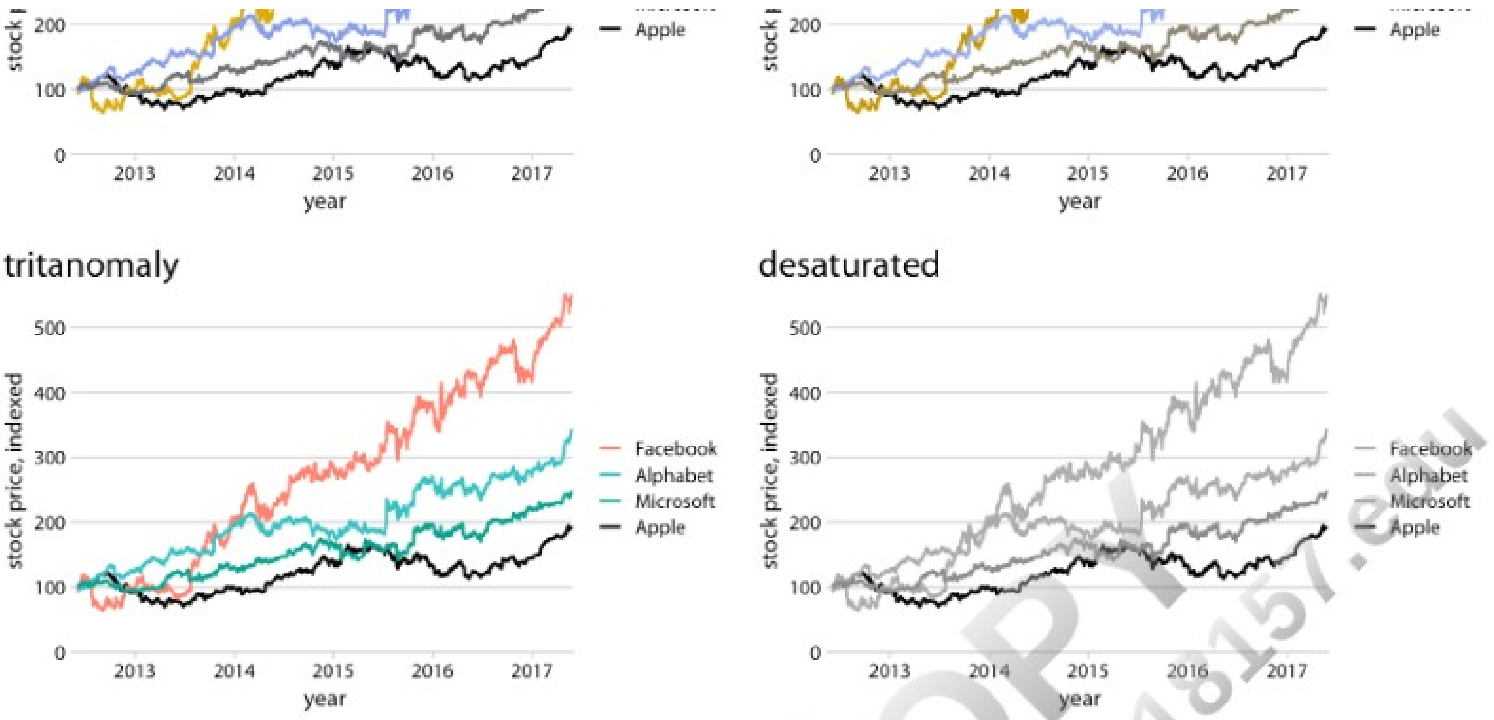


Figure 20-7. Color-vision deficiency simulation of Figure 20-6. Data source: Yahoo! Finance.

Designing Figures Without Legends

Even though legend legibility can be improved by encoding data redundantly, in multiple aesthetics, legends always put an extra mental burden on the reader. In reading a legend, the reader needs to pick up information in one part of the visualization and then transfer it over to a different part. We can typically make our readers' lives easier if we eliminate the legend altogether. Eliminating the legend does not mean, however, that we simply don't provide one and instead write sentences such as "The yellow dots represent *Iris versicolor*" in the figure caption. Eliminating the legend means that we design the figure in such a way that it is immediately obvious what the various graphical elements represent, even if no explicit legend is present.

The general strategy we can employ is called *direct labeling*, whereby we incorporate appropriate text labels or other visual elements that serve as guideposts to the rest of the figure. We have previously encountered direct labeling in Chapter 19 (Figure 19-2), as an alternative to drawing a legend with over 50 distinct colors. To apply the direct labeling concept to the stock price figure, we place the name of each company right next to the end of its respective data line (Figure 20-8).

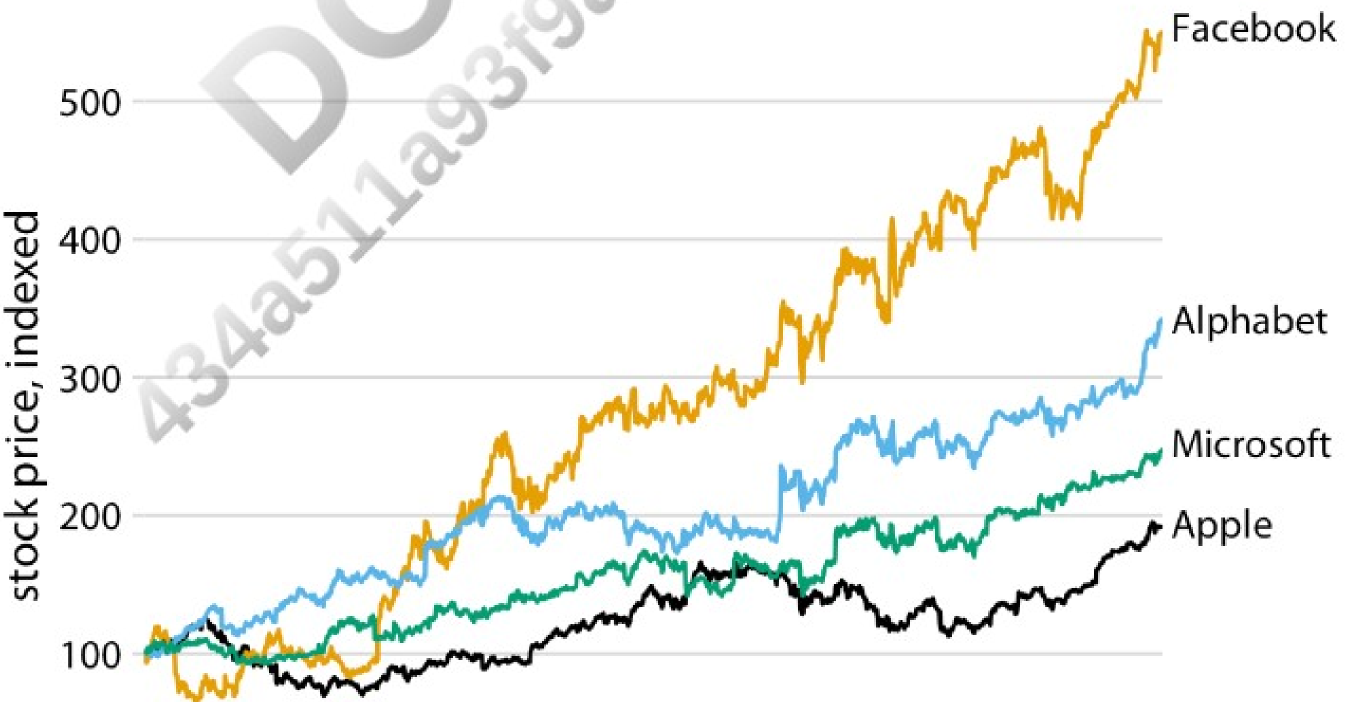
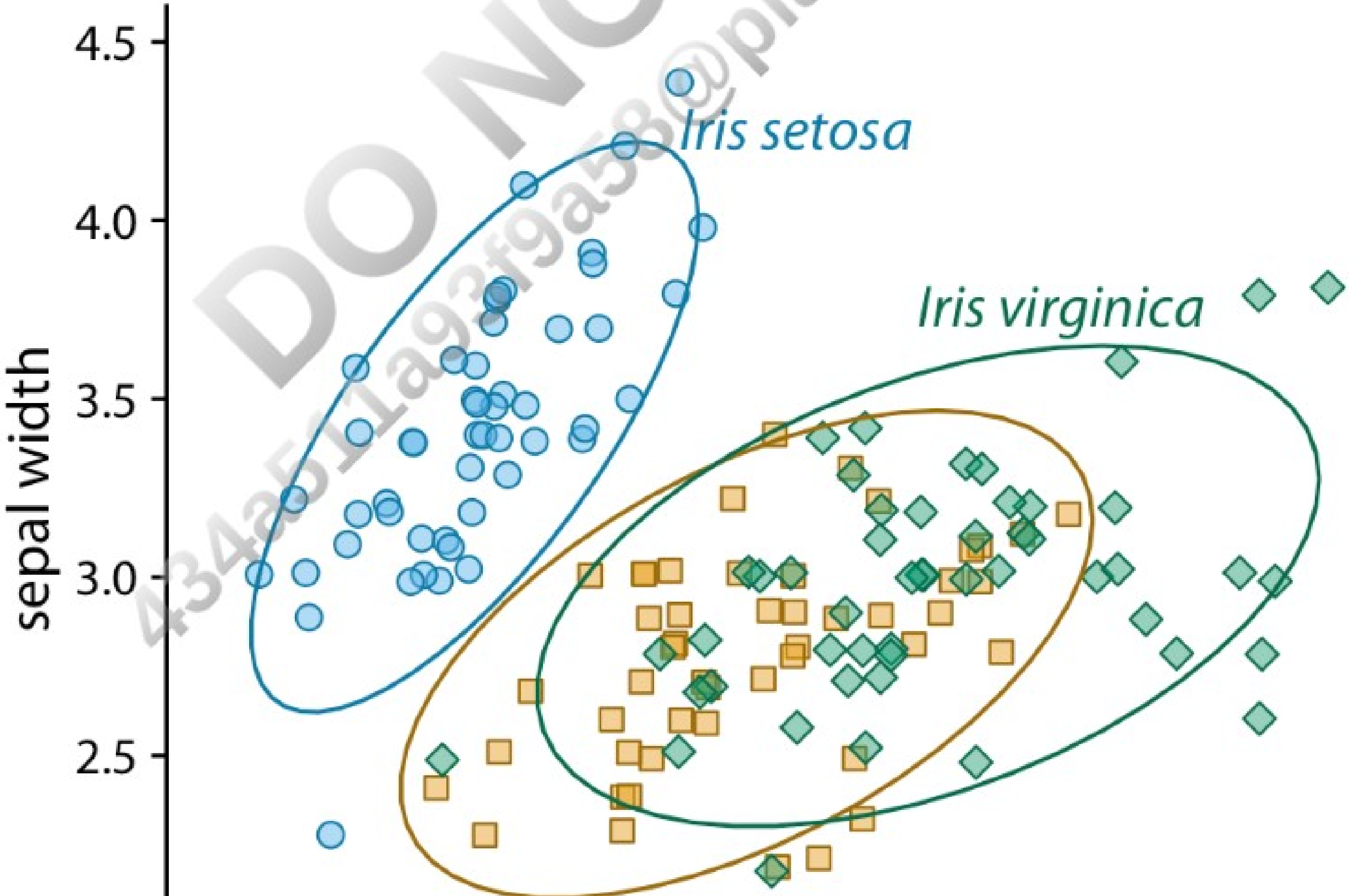


Figure 20-8. Stock price over time for four major tech companies. The stock price for each company has been normalized to equal 100 in June 2012. Data source: Yahoo! Finance.

> **TIP**
>
> Whenever possible, design your figures so they don't need a separate legend.

We can also apply the direct labeling concept to the *Iris* data from the beginning of this chapter, specifically Figure 20-3. Because it is a scatterplot of many points that separate into three different groups, we need to directly label the groups rather than the individual points. One solution is to draw ellipses that enclose the majority of the points and then label the ellipses (Figure 20-9).

For density plots, we can similarly direct-label the curves rather than providing a color-coded legend (Figure 20-10). In both Figures 20-9 and 20-10, I have colored the text labels in the same colors as the data. Colored labels can greatly enhance the direct labeling effect, but they can also turn out poorly. If the text labels are printed in a color that is too light, then the labels become difficult to read. And because text consists of very thin lines, colored text often appears to be lighter than an adjacent filled area of the same color. I generally circumvent these issues by using two different shades of each color, a light one for filled areas and a dark one for lines, outlines, and text. If you carefully inspect Figure 20-9 or 20-10, you will see how each data point or shaded area is filled with a light color and has an outline drawn in a darker color of the same hue. The text labels are drawn in the same darker colors.



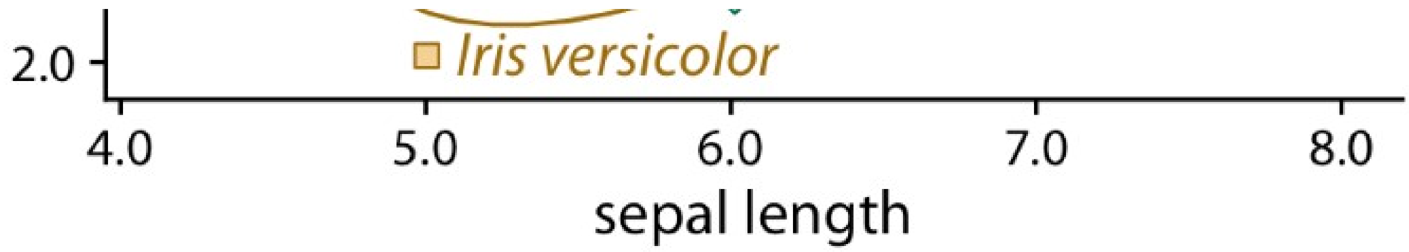


Figure 20-9. Sepal width versus sepal length for three different Iris species. The points representing different Iris species have been directly labeled with colored ellipses and text labels. Compared to Figure 20-3, I have removed the background grid here because the figure was becoming too busy. Data source: [Fisher 1936].

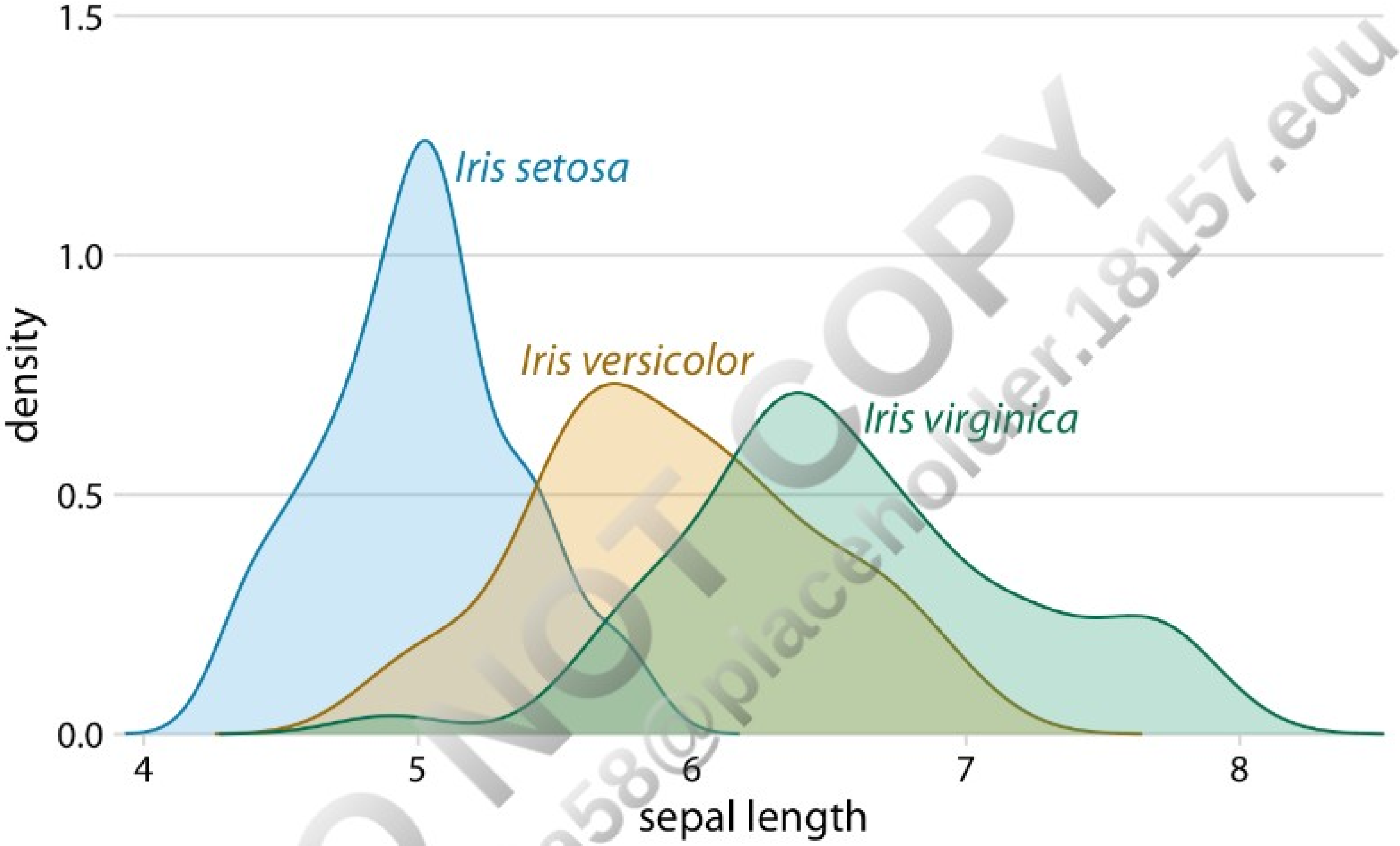
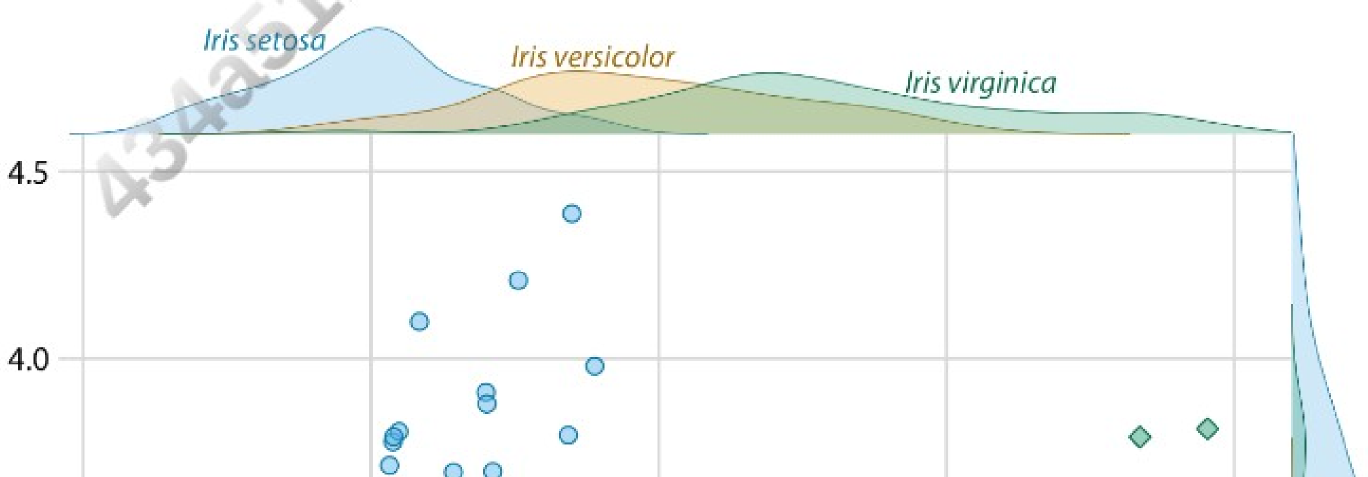


Figure 20-10. Density estimates of the sepal lengths of three different Iris species. Each density estimate is directly labeled with the respective species name. Data source: [Fisher 1936].

We can also use density plots such as the one in Figure 20-10 as a legend replacement, by placing the density plots into the margins of a scatterplot (Figure 20-11). This allows us to direct-label the marginal density plots rather than the central scatterplot and hence results in a figure that is somewhat less cluttered than Figure 20-9 with its directly labeled ellipses.



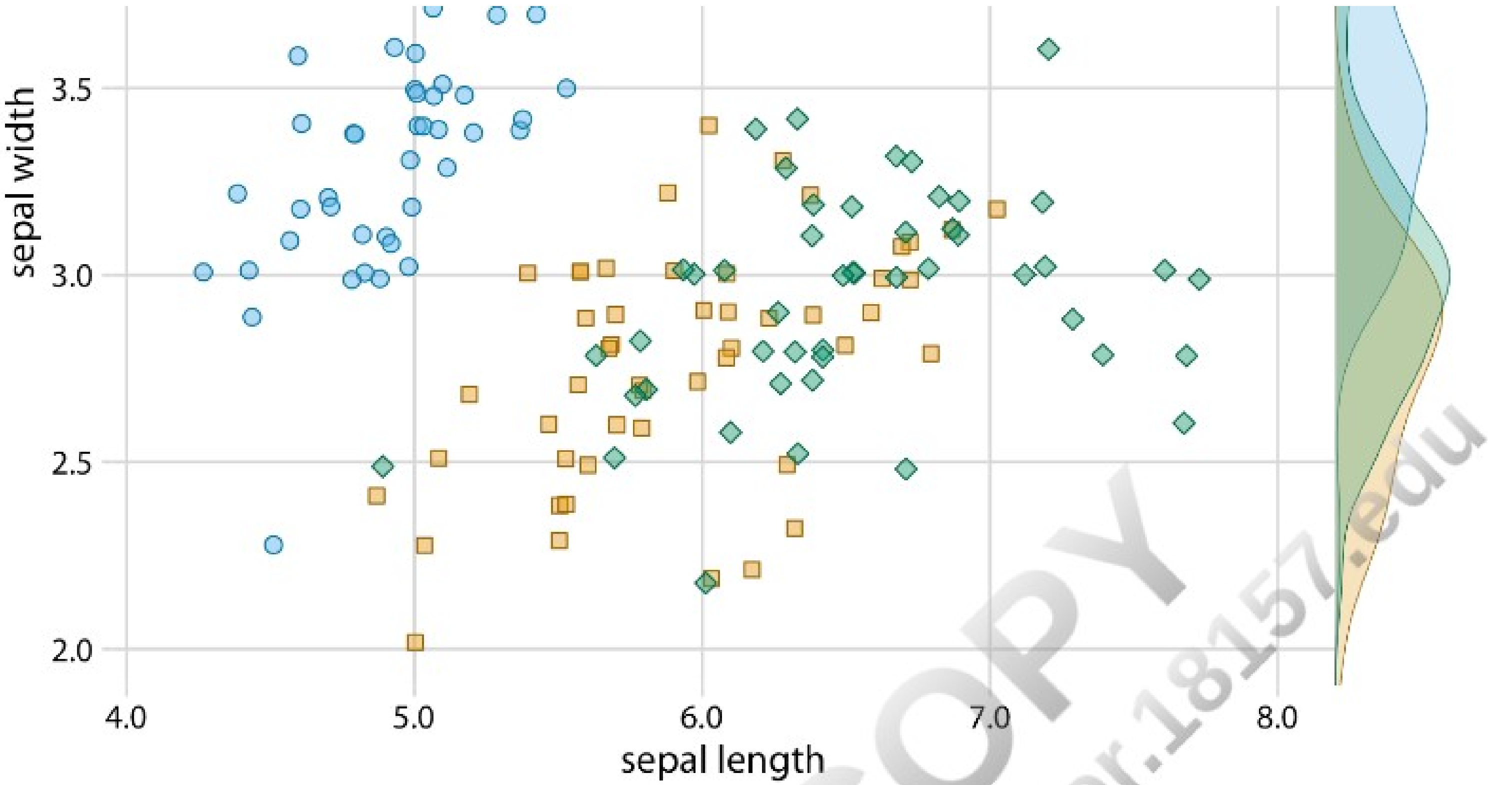


Figure 20-11. Sepal width versus sepal length for three different Iris species, with marginal density estimates of each variable for each species. Data source: [Fisher 1936].

And finally, whenever we encode a single variable in multiple aesthetics, we don’t normally want multiple separate legends for the different aesthetics. Instead, there should be a single legend-like visual element that conveys all the mappings at once. In the case where we map the same variable onto a position along a major axis and onto color, this implies that the reference color bar should run along and be integrated into the same axis. Figure 20-12 shows a case where we map temperature to both a position along the *x* axis and to color, and where we therefore have integrated the color legend into the *x* axis.

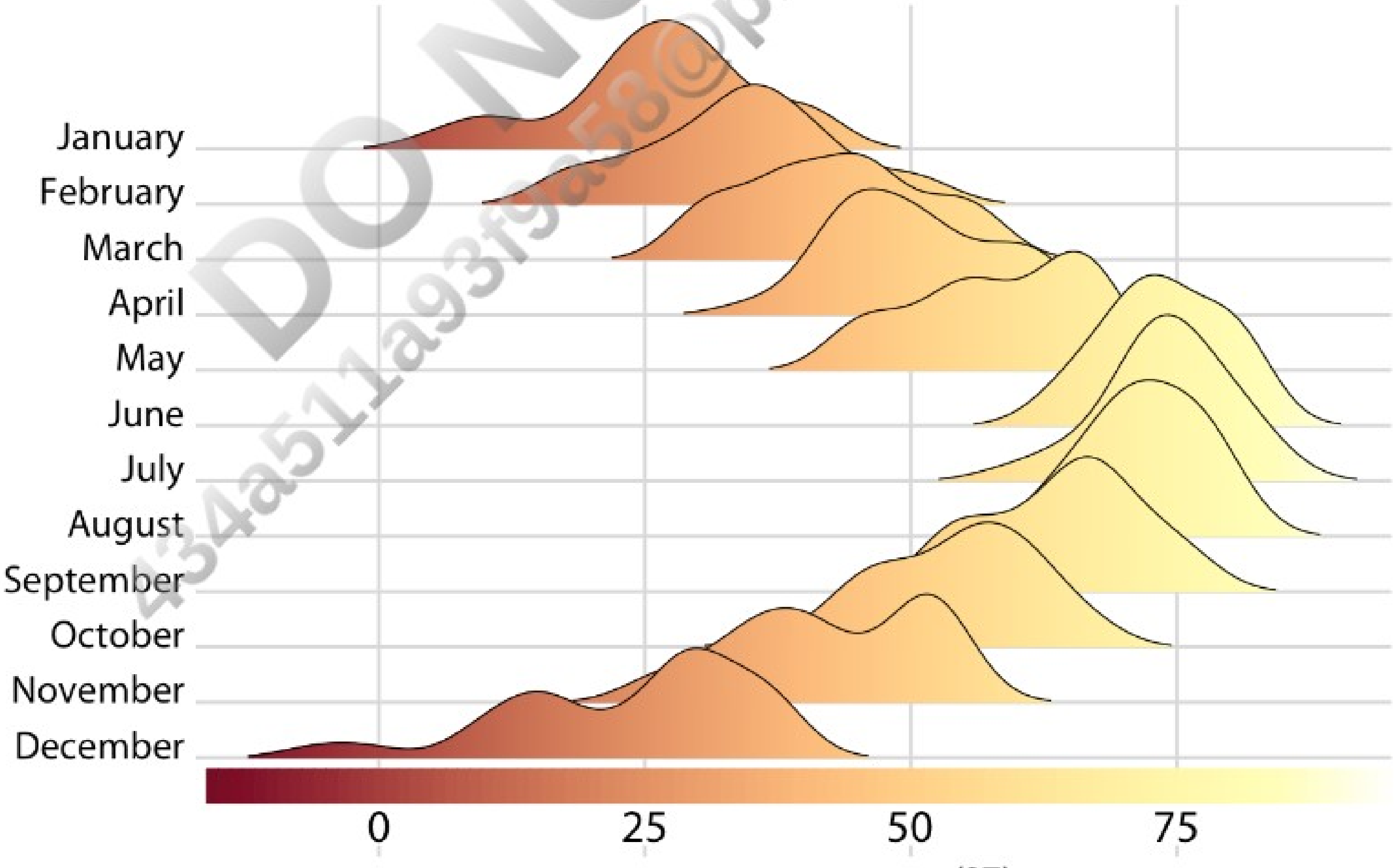


Figure 20-12. Temperatures in Lincoln, NE, in 2016. This figure is a variation of Figure 9-9. Temperature is now shown both by location along the x axis and by color, and a color bar along the x axis visualizes the scale that converts temperatures into colors. Data source: Weather Underground.