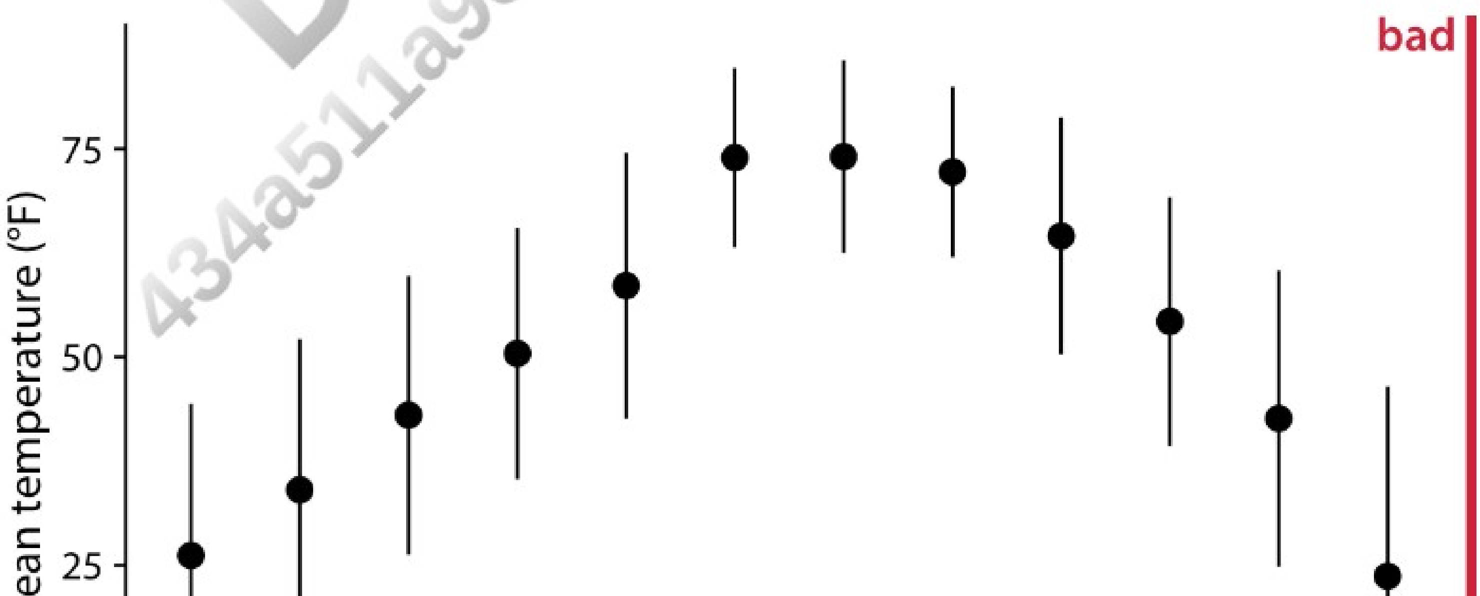
Chapter 9. Visualizing Many Distributions at Once

There are many scenarios in which we want to visualize multiple distributions at the same time. For example, consider weather data. We may want to visualize how temperature varies across different months while also showing the distribution of observed temperatures within each month. This scenario requires showing a dozen temperature distributions at once, one for each month. None of the visualizations discussed in Chapters 7 or 8 work well in this case. Instead, viable approaches include boxplots, violin plots, and ridgeline plots.

Whenever we are dealing with many distributions, it is helpful to think in terms of the response variable and one or more grouping variables. The *response variable* is the variable whose distributions we want to show. The *grouping variables* define subsets of the data with distinct distributions of the response variable. For example, for temperature distributions across months, the response variable is the temperature and the grouping variable is the month. All techniques discussed in this chapter draw the response variable along one axis and the grouping variable(s) along the other. In the following sections, I will first describe approaches that show the response variable along the vertical axis, and then I will describe approaches that show the response variable along the horizontal axis. In all cases discussed, we could flip the axes and arrive at an alternative and viable visualization. I am showing here the canonical forms of the various visualizations.

Visualizing Distributions Along the Vertical Axis

The simplest approach to showing many distributions at once is to show their mean or median as points, with some indication of the variation around the mean or median shown by error bars. Figure 9-1 demonstrates this approach for the distributions of monthly temperatures in Lincoln, Nebraska, in 2016. I have labeled this figure as “bad” because there are multiple problems with this approach. First, by representing each distribution by only one point and two error bars, we are losing a lot of information about the data. Second, it is not immediately obvious what the points represent, even though most readers would likely guess that they represent either the mean or the median. Third, it is definitely not obvious what the error bars represent. Do they represent the standard deviation of the data, the standard error of the mean, a 95% confidence interval, or something else altogether? There is no commonly accepted standard. By reading the figure caption of Figure 9-1, we can see that they represent here twice the standard deviation of the daily mean temperatures, meant to indicate the range that contains approximately 95% of the data. However, error bars are more commonly employed to visualize the standard error (or twice the standard error for a 95% confidence interval), and it is easy for readers to confuse the standard error with the standard deviation. The standard error quantifies how accurate our estimate of the mean is, whereas the standard deviation estimates how much spread there is in the data around the mean. It is possible for a dataset to have both a very small standard error of the mean and a very large standard deviation. Fourth, symmetric error bars are misleading if there is any skew in the data, which is the case here and is almost always for real-world datasets.



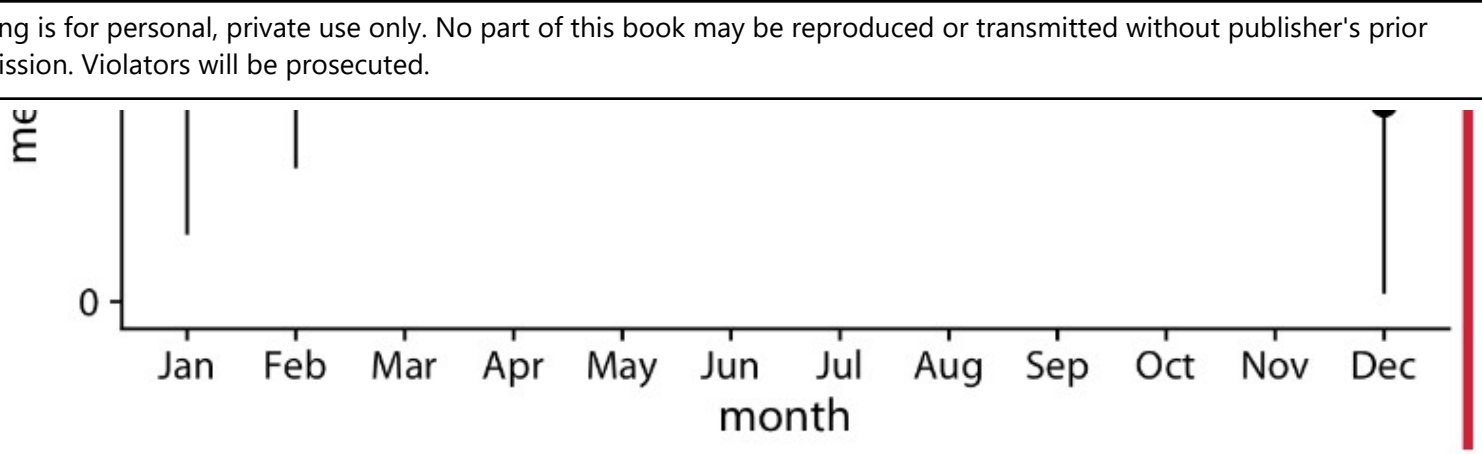


Figure 9-1. Mean daily temperatures in Lincoln, NE, in 2016. Points represent the average daily mean temperatures for each month, averaged over all days of the month, and error bars represent twice the standard deviation of the daily mean temperatures within each month. This figure has been labeled as "bad" because error bars are conventionally used to visualize the uncertainty of an estimate, not the variability in a population. Data source: Weather Underground.

We can address all four shortcomings of Figure 9-1 by using a traditional and commonly used method for visualizing distributions, the *boxplot*. A boxplot divides the data into quartiles and visualizes them in a standardized manner (Figure 9-2).

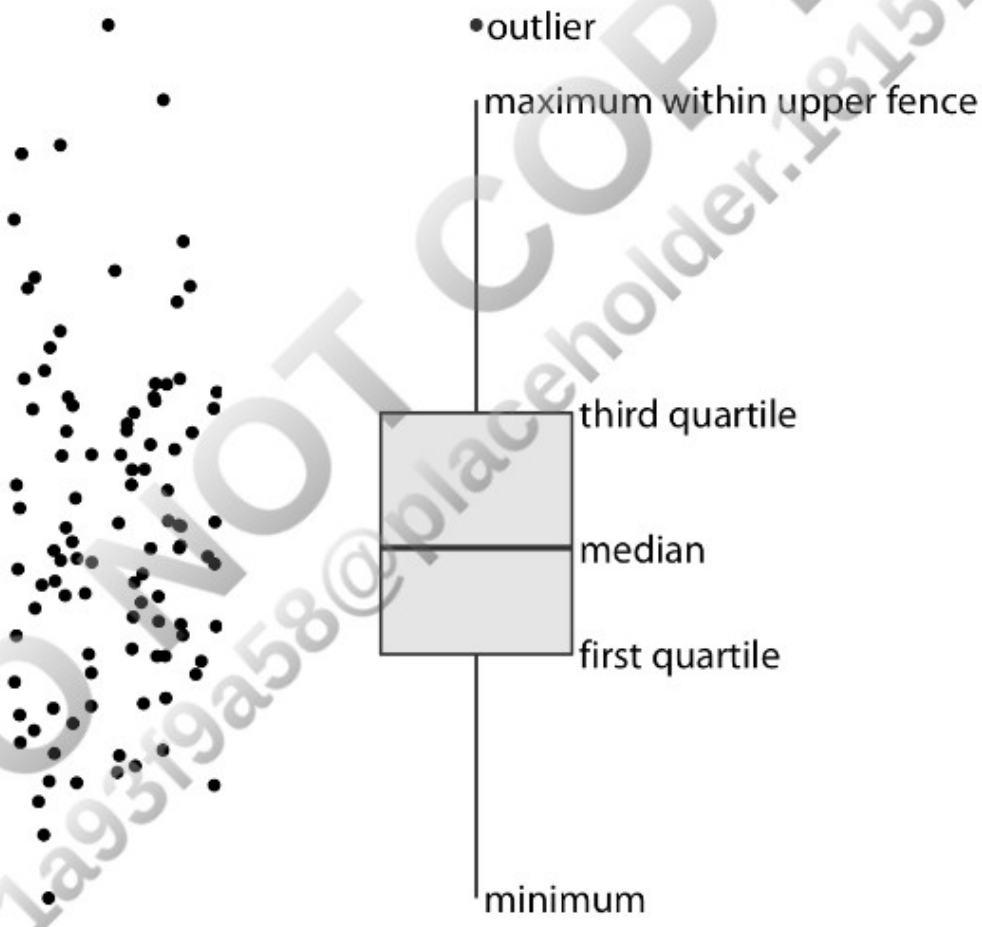


Figure 9-2. Anatomy of a boxplot. Shown are a cloud of points (left) and the corresponding boxplot (right).

Only the *y* values of the points are visualized in the boxplot in Figure 9-2. The line in the middle of the boxplot represents the median, and the box encloses the middle 50% of the data. The vertical lines extending upwards and downwards from the box are called *whiskers*. The top and bottom whiskers extend either to the maximum and minimum values of the data or to the maximum or minimum values that fall within 1.5 times the height of the box, whichever yields the shorter whisker. The distances of 1.5 times the height of the box in either direction are called the upper and lower *fences*. Individual data points that fall beyond the fences are referred to as outliers and are usually shown as individual dots.

Boxplots are simple yet informative, and they work well when plotted next to each other to visualize many distributions at once. For the Lincoln temperature data, using boxplots leads to Figure 9-3. In that figure, we can now see that temperature is highly skewed in December (most days are moderately cold and a few are extremely cold) and not very skewed at all in some other months, such as in July.

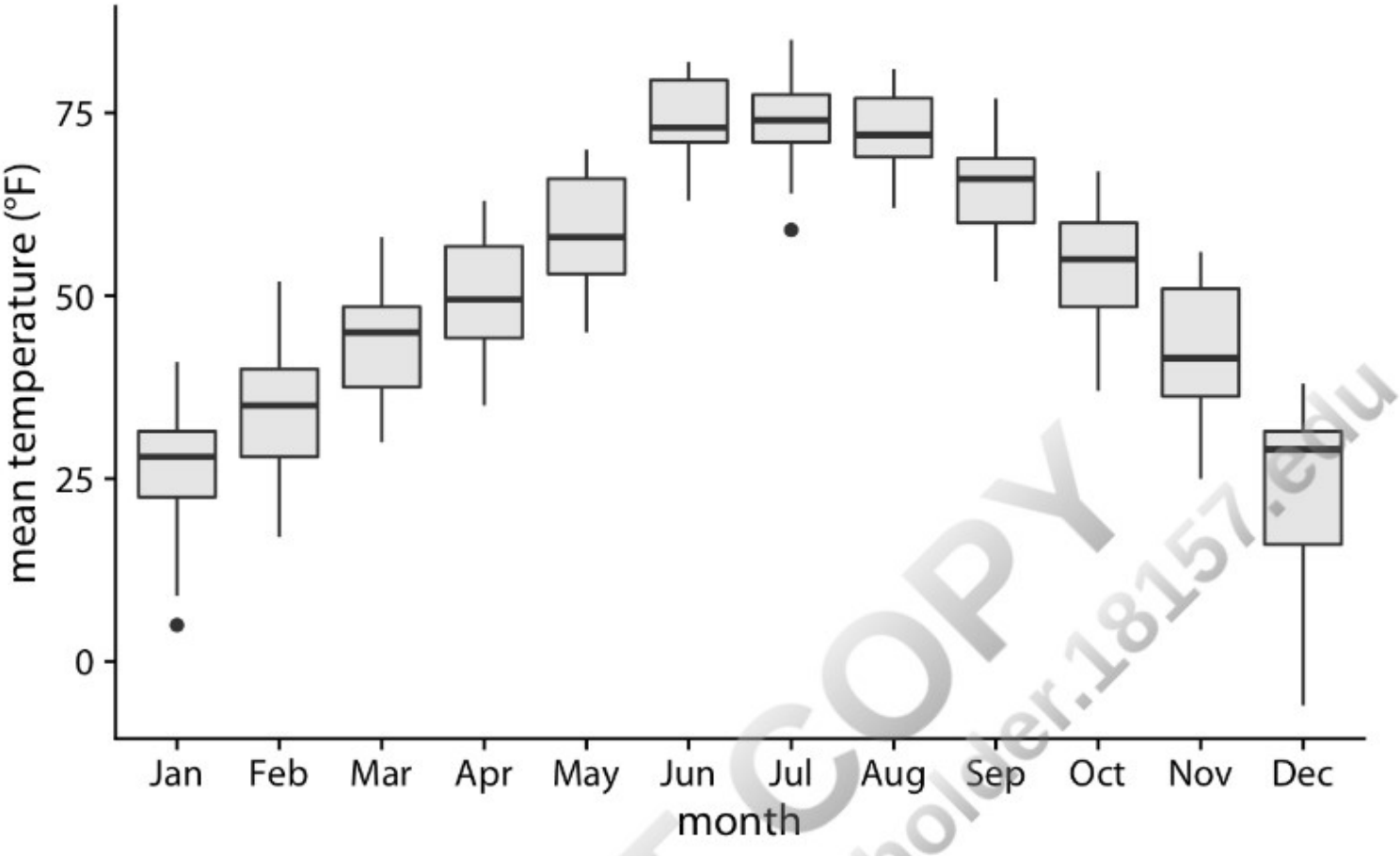
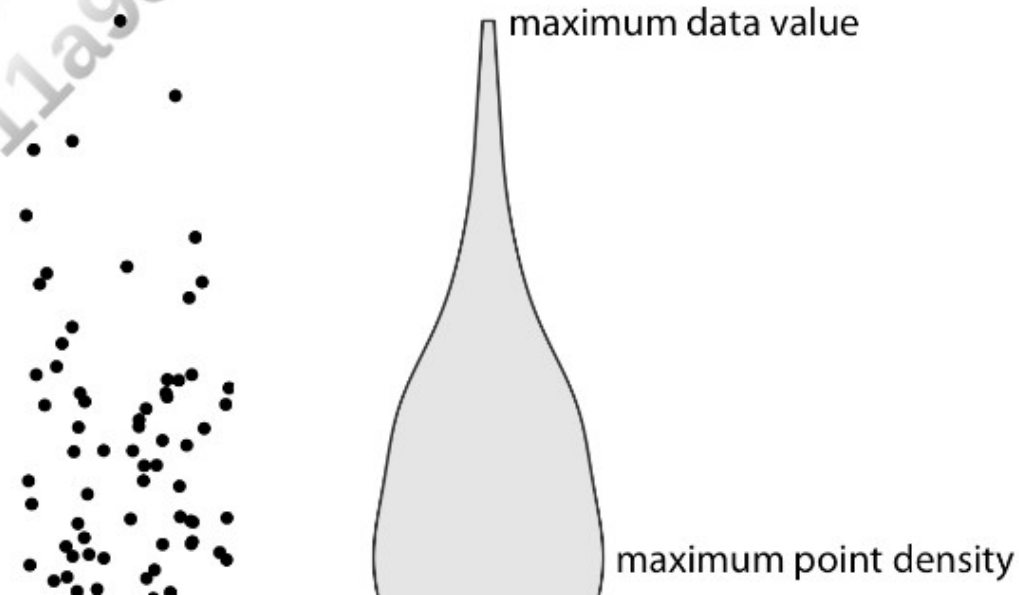


Figure 9-3. Mean daily temperatures in Lincoln, NE, visualized as boxplots. Data source: Weather Underground.

Boxplots were invented by the statistician John Tukey in the early 1970s, and they quickly gained popularity because they were highly informative while being easy to draw by hand, which is how most data visualizations were drawn at that time. However, with modern computing and visualization capabilities, we are not limited to what is easily drawn by hand. Therefore, more recently we see boxplots being replaced by *violin plots* (Figure 9-4). Violins can be used whenever one would otherwise use a boxplot, and they provide a much more nuanced picture of the data. In particular, violin plots will accurately represent bimodal data whereas a boxplot will not.

Only the *y* values of the points are visualized in the violin plot. The width of the violin at a given *y* value represents the point density at that *y* value. Technically, a violin plot is a density estimate rotated by 90 degrees and then mirrored (Chapter 7). Violins are therefore symmetric. Violins begin and end at the minimum and maximum data values, respectively. The thickest part of the violin corresponds to the highest point density in the dataset.



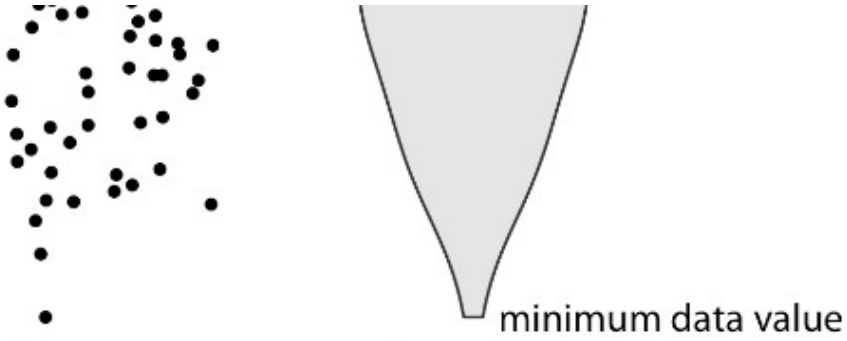


Figure 9-4. Anatomy of a violin plot. Shown are a cloud of points (left) and the corresponding violin plot (right).

> **NOTE**
>
> Before using violins to visualize distributions, verify that you have sufficiently many data points in each group to justify showing the point densities as smooth lines.

When we visualize the Lincoln temperature data with violins, we obtain Figure 9-5. We can now see that some months do have moderately bimodal data. For example, the month of November seems to have had two temperature clusters, one around 50 degrees and one around 35 degrees Fahrenheit.

Because violin plots are derived from density estimates, they have similar shortcomings. In particular, they can generate the appearance that there is data where none exists, or that the dataset is very dense when actually it is quite sparse. We can try to circumvent these issues by simply plotting all the individual data points directly, as dots (Figure 9-6). Such a figure is called a *strip chart*. Strip charts are fine in principle, as long as we make sure that we don't plot too many points on top of each other. A simple solution to overplotting is to spread out the points somewhat along the *x* axis, by adding some random noise in the *x* dimension (Figure 9-7). This technique is called *jittering*.

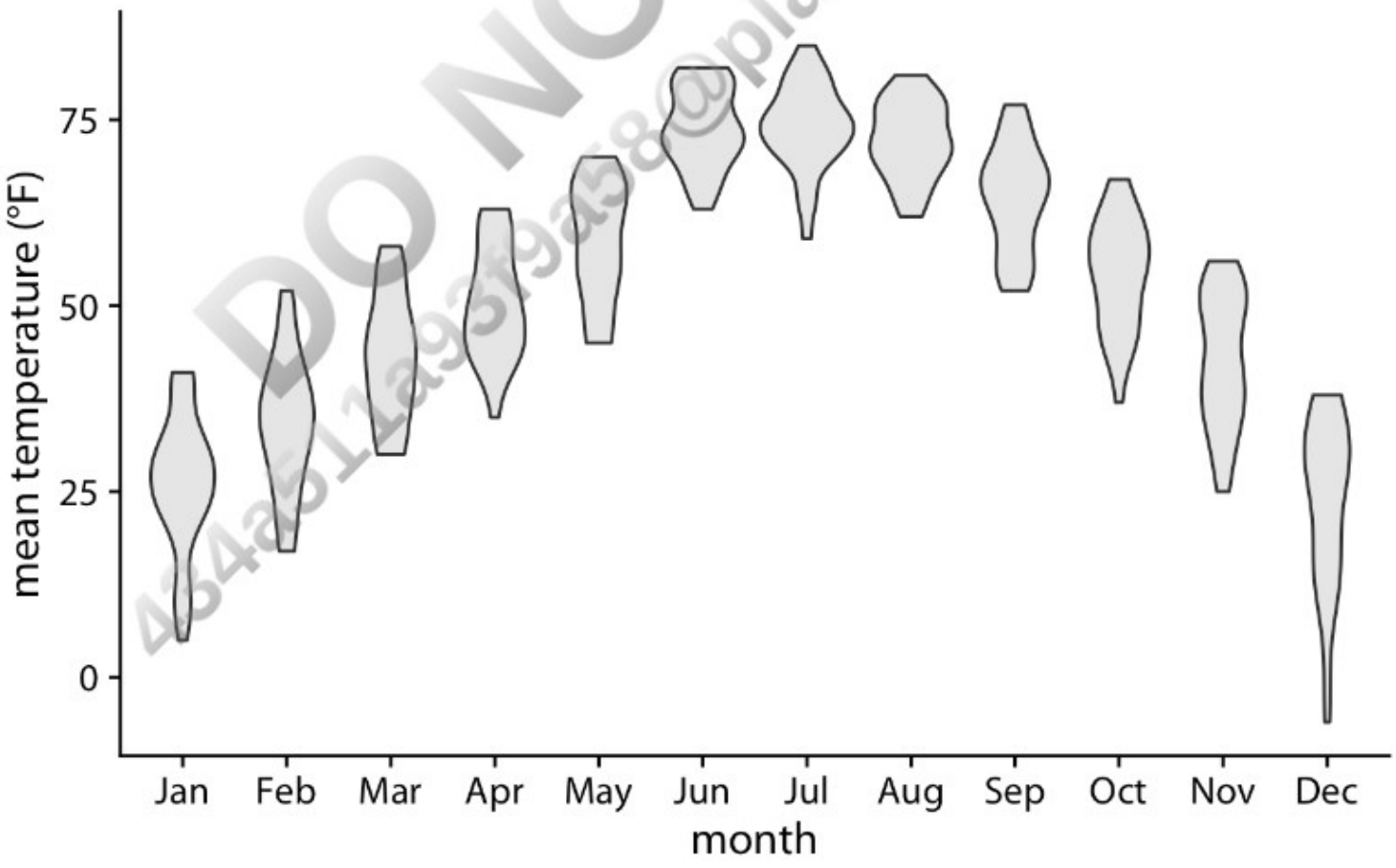


Figure 9-5. Mean daily temperatures in Lincoln, NE, visualized as violin plots. Data source: Weather Underground.

Figure 9-5. Mean daily temperatures in Lincoln, NE, visualized as violin plots. Data source: Weather Underground.

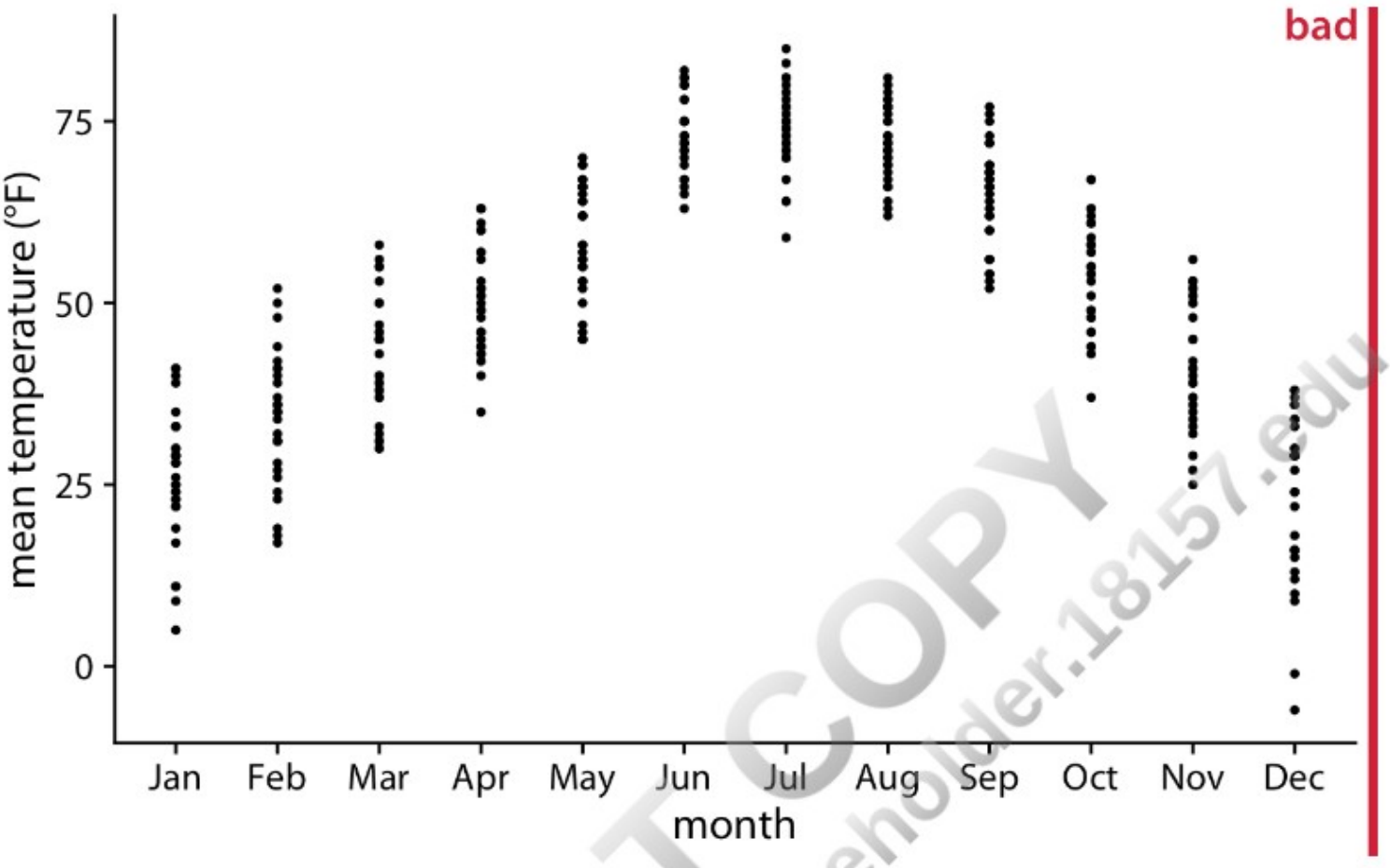


Figure 9-6. Mean daily temperatures in Lincoln, NE, visualized as strip charts. Each point represents the mean temperature for one day. This figure is labeled as "bad" because so many points are plotted on top of each other that it is not possible to ascertain which temperatures were the most common in each month. Data source: Weather Underground.

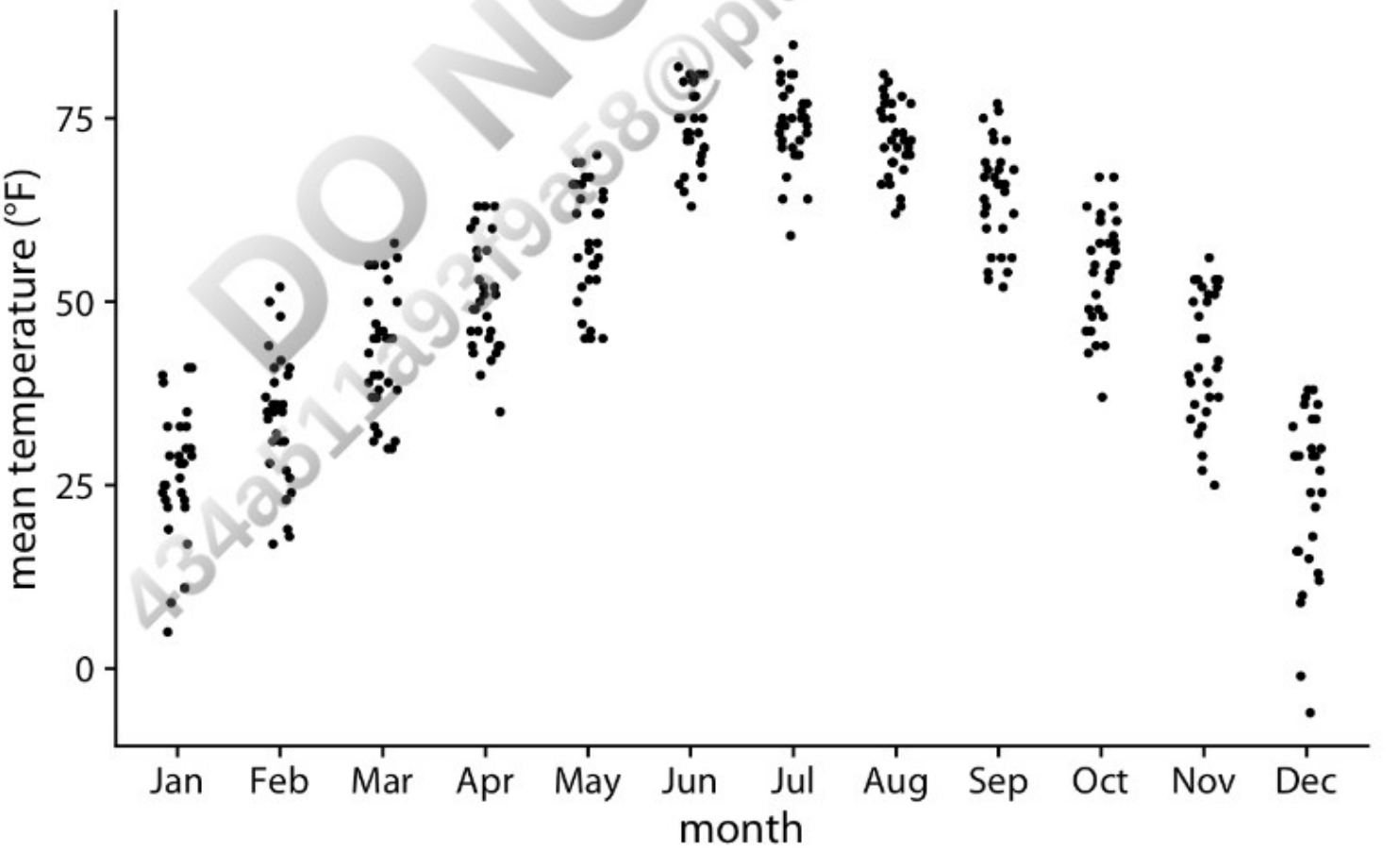


Figure 9-7. Mean daily temperatures in Lincoln, NE, visualized as strip charts. The points have been jittered along the x axis to better show the density of points at each temperature value. Data source: Weather Underground.

> **NOTE**
>
> Whenever the dataset is too sparse to justify the violin visualization, plotting the raw data as individual points will be possible.

Finally, we can combine the best of both worlds by spreading out the dots in proportion to the point density at a given *y* coordinate. This method, called a *sina plot* [Sidiropoulos et al. 2018],[1] can be thought of as a hybrid between a violin plot and jittered points, and it shows each individual point while also visualizing the distributions. In Figure 9-8, I have drawn the sina plots on top of the violins to highlight the relationship between these two approaches.

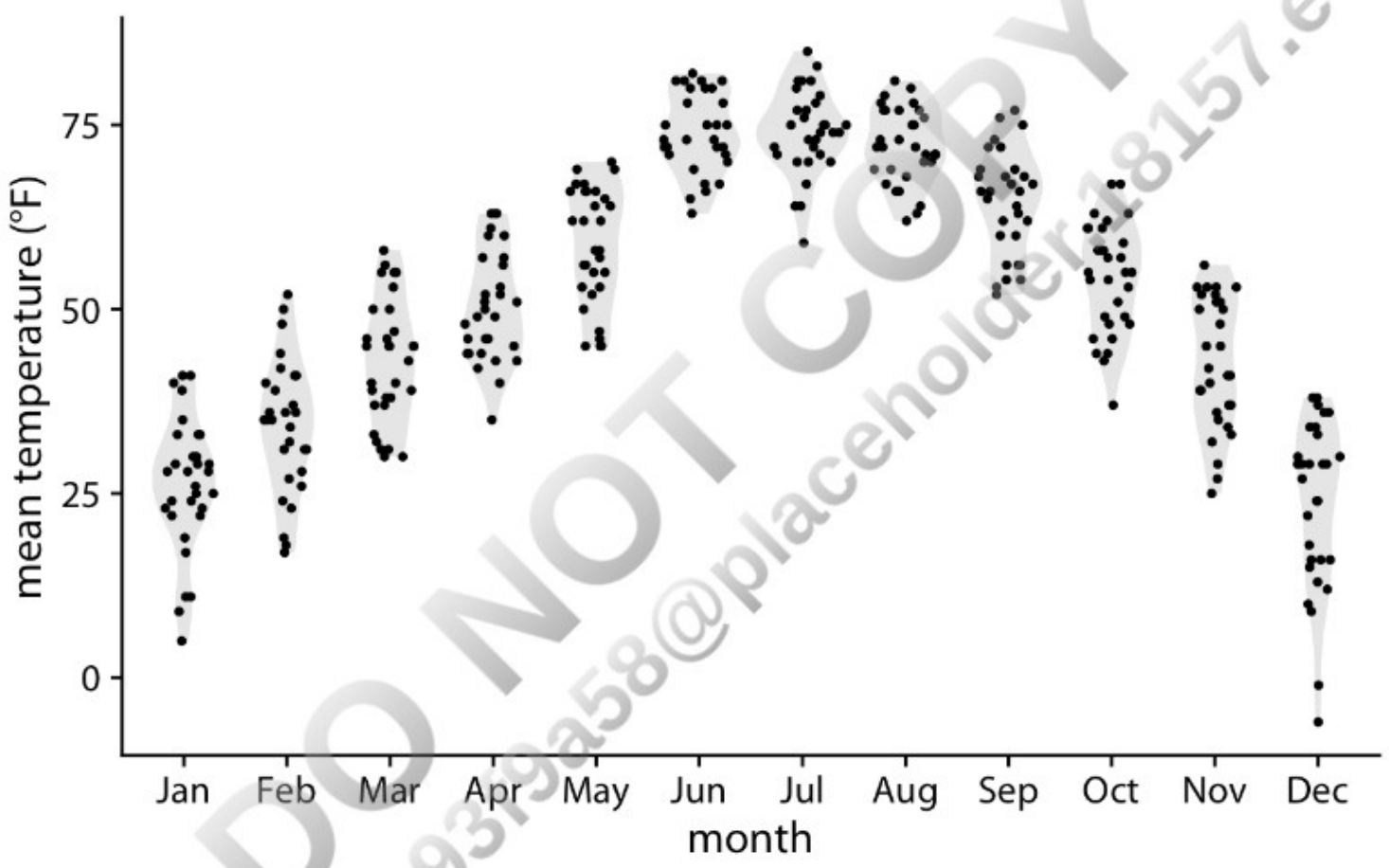


Figure 9-8. Mean daily temperatures in Lincoln, NE, visualized as sina plots (a combination of individual points and violins). The points have been jittered along the x axis in proportion to the point density at the respective temperature. Here, the sina plots are shown superimposed on violin plots. Data source: Weather Underground.

Visualizing Distributions Along the Horizontal Axis

In Chapter 7, we visualized distributions along the horizontal axis using histograms and density plots. Here, we will expand on this idea by staggering the distribution plots in the vertical direction. The resulting visualization is called a *ridgeline plot*, because these plots look like mountain ridgelines. Ridgeline plots tend to work particularly well if you want to show trends in distributions over time.

The standard ridgeline plot uses density estimates (Figure 9-9). It is quite closely related to the violin plot, but frequently evokes a more intuitive understanding of the data. For example, the two clusters of temperatures around 35 degrees and 50 degrees Fahrenheit in November are much more obvious in Figure 9-9 than in Figure 9-5.

Because the *x* axis shows the response variable and the *y* axis shows the grouping variable, there is no separate axis for the

density estimates in a ridgeline plot. Density estimates are shown alongside the grouping variable. This is no different from the violin plot, where densities are also shown alongside the grouping variable, without a separate, explicit scale. In both cases, the purpose of the plot is not to show specific density values but instead to allow for easy comparison of density shapes and relative heights across groups.

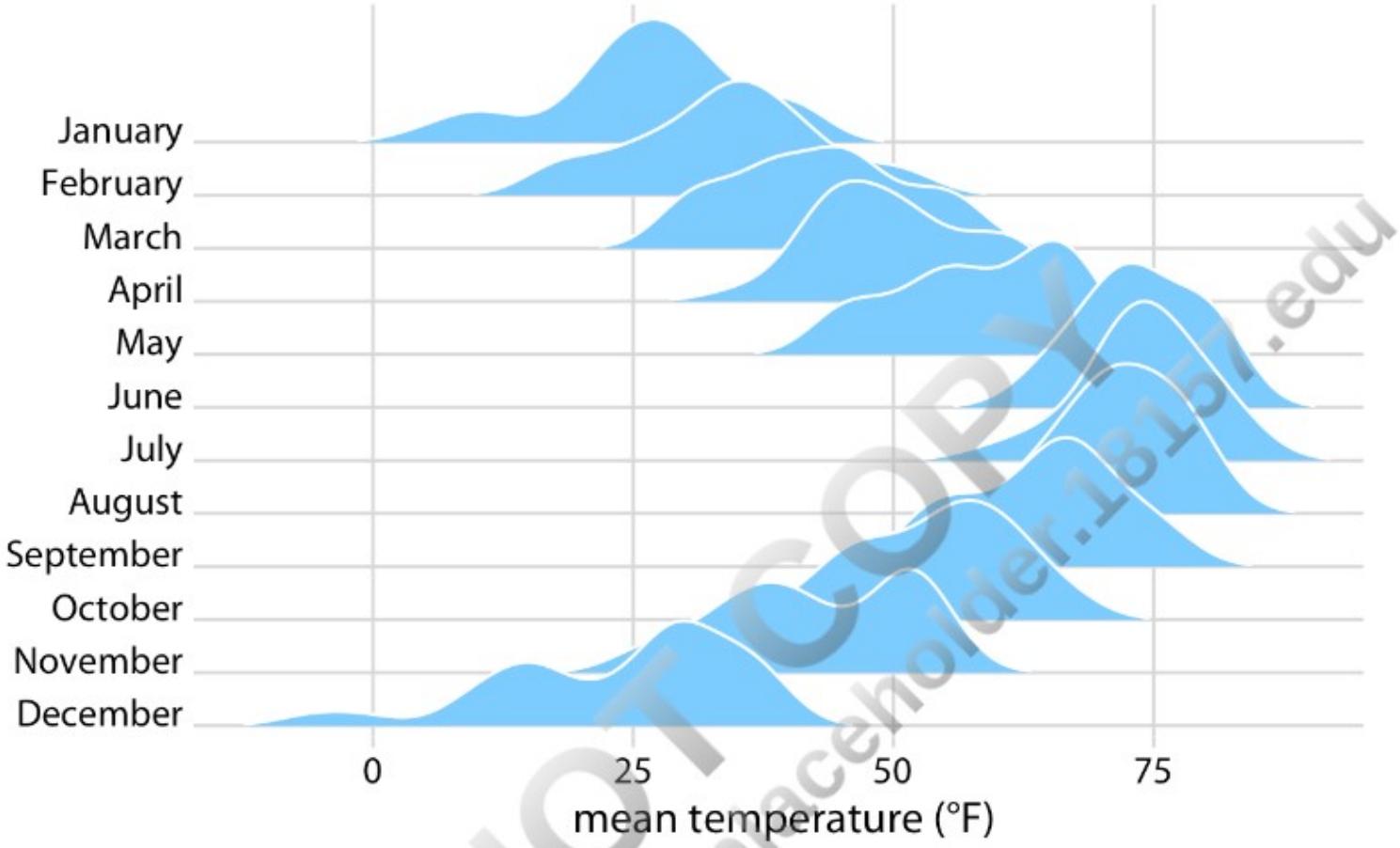


Figure 9-9. Temperatures in Lincoln, NE, in 2016, visualized as a ridgeline plot. For each month, we show the distribution of daily mean temperatures measured in Fahrenheit. Original figure concept: [Wehrwein 2017]. Data source: Weather Underground.

In principle, we can use histograms instead of density plots in a ridgeline visualization. However, the resulting figures often don't look very good (Figure 9-10). The problems are similar to those of stacked or overlapping histograms (see "Visualizing Multiple Distributions at the Same Time"). Because the vertical lines in these ridgeline histograms always appear at the exact same *x*

x values, the bars from different histograms align with each other in confusing ways. In my opinion, it is better to not draw such overlapping histograms.

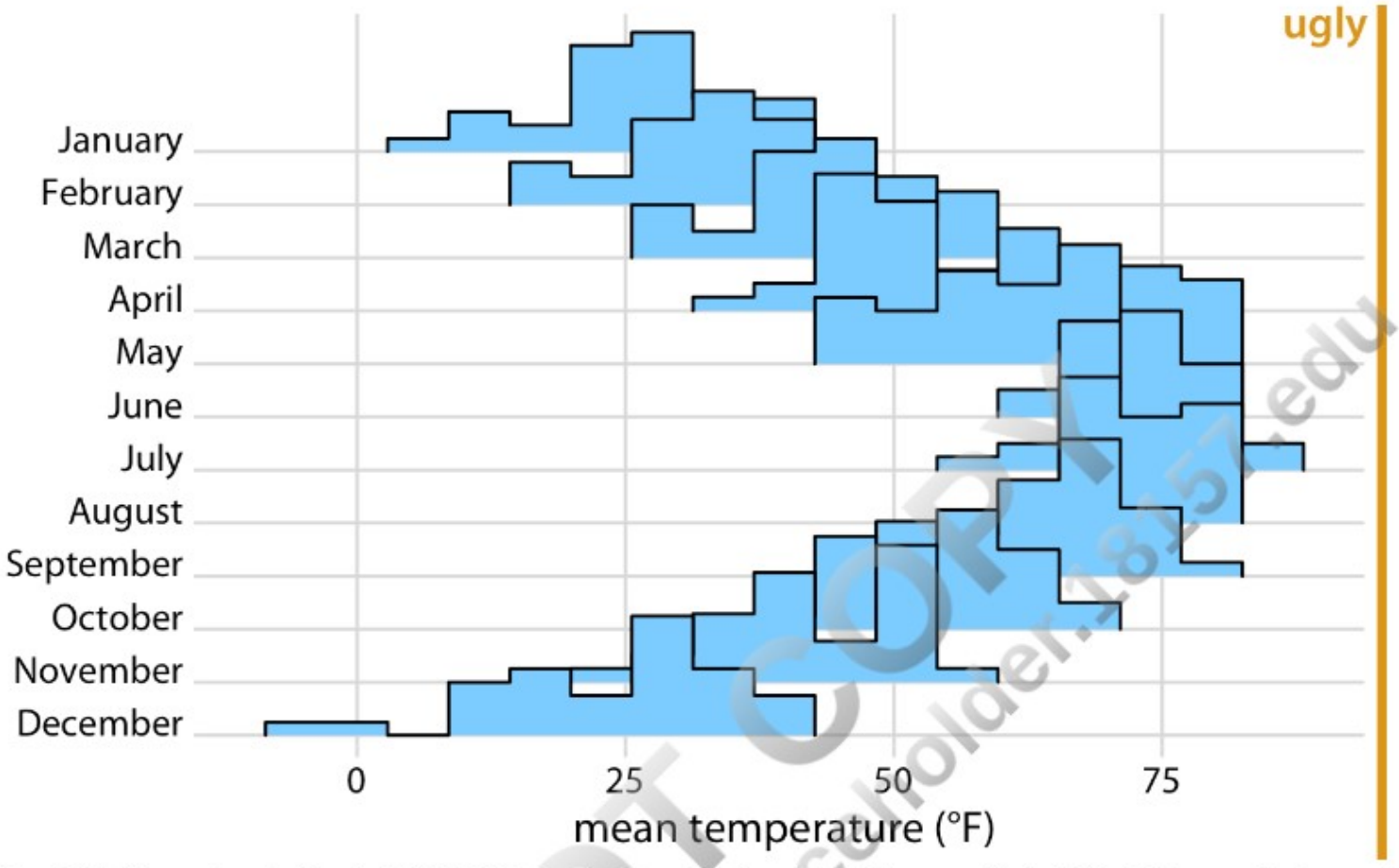
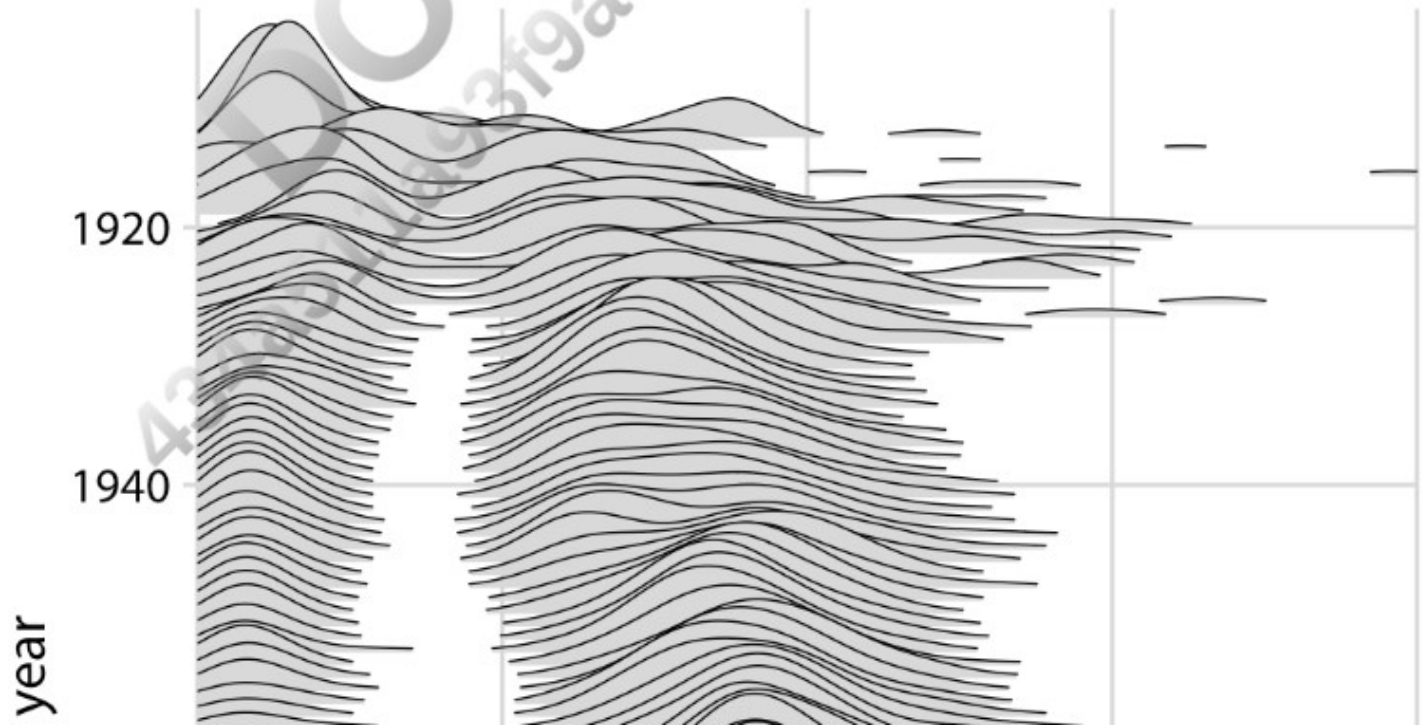


Figure 9-10. Temperatures in Lincoln, NE, in 2016, visualized as a ridgeline plot of histograms. The individual histograms don't separate well visually, and the overall figure is quite busy and confusing. Data source: Weather Underground.

Ridgeline plots scale to very large numbers of distributions. For example, Figure 9-11 shows the distributions of movie lengths from 1913 to 2005. This figure contains almost 100 distinct distributions and yet it is very easy to read. We can see that in the 1920s, movies came in many different lengths, but since about 1960 movie length has standardized to approximately 90 minutes.



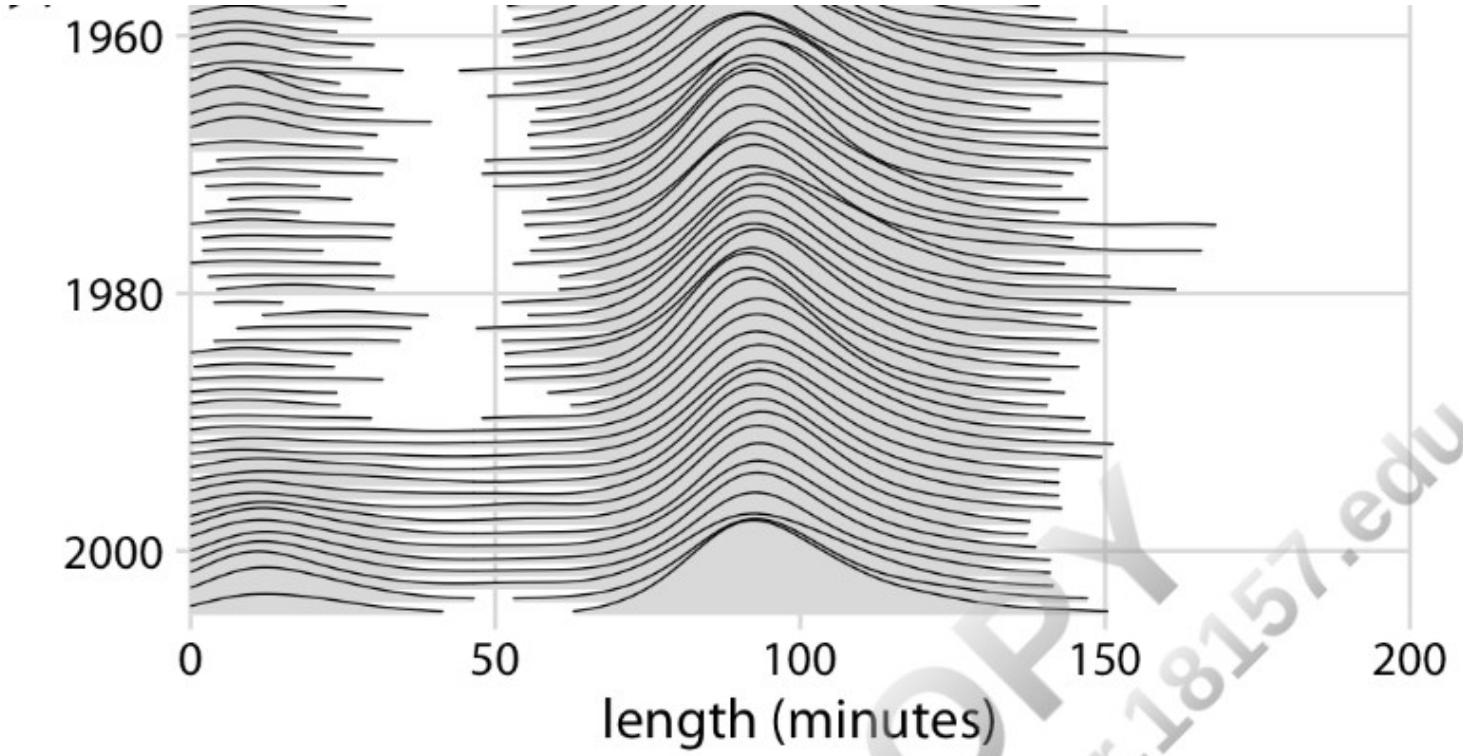


Figure 9-11. Evolution of movie lengths over time. Since the 1960s, the majority of all movies have been approximately 90 minutes long. Data source: Internet Movie Database (IMDB).

Ridgeline plots also work well if we want to compare two trends over time. This is a scenario that arises commonly if we want to analyze the voting patterns of the members of two different parties. We can make this comparison by staggering the distributions vertically by time and drawing two differently colored distributions at each time point, representing the two parties (Figure 9-12).

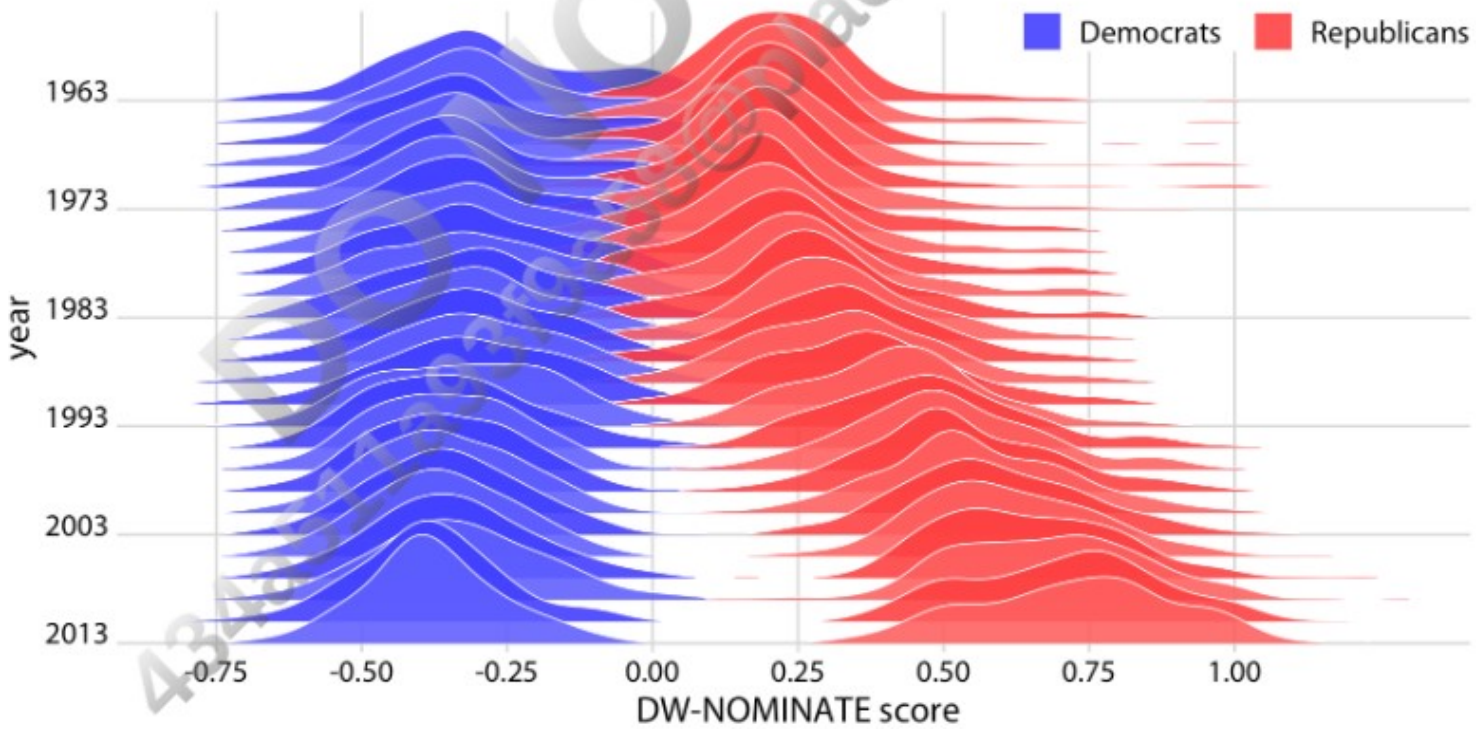


Figure 9-12. Voting patterns in the US House of Representatives have become increasingly polarized. DW-NOMINATE scores are frequently used to compare voting patterns of representatives between parties and over time. Here, score distributions are shown for each Congress from 1963 to 2013 separately for Democrats and Republicans. Each Congress is represented by its first year. Original figure concept: [McDonald 2017]. Data source: Keith Poole.

---

1 The name *sina plot* is meant to honor Sina Hadi Sohi, a student at the University of Copenhagen, Denmark, who wrote the first

[1] The name *Sina plot* is meant to honor Sina Hadi Sohi, a student at the University of Copenhagen, Denmark, who wrote the first version of the code that researchers at the university used to make such plots (Frederik O. Bagger, personal communication).