

## Chapter 2

# Framework

*“The time has come,’ the Walrus said,  
“To talk of many things:  
Of shoes — and ships — and sealing-wax —  
Of cabbages — and kings —  
And why the sea is boiling hot —  
And whether pigs have wings.”*

— Lewis Carroll, *Through the Looking-Glass and  
What Alice Found There* (1872)

### 2.1 How to Speak Visualization

In the *Survey of English Dialects*,<sup>1</sup> Dieth and Orton [84] explored how different words were used for the same objects in various areas of England. The variety of words is substantial; the place where a farmer might keep his cows is called a *byre*, a *shippon*, a *mistall*, a *cow-stable*, a *cow-house*, a *cow-shed*, a *neat-house*, or a *beast-house*. Perhaps, then, it is not so surprising that we see the same situation in visualization, where a 2-D chart with data displayed as a collection of points, using one variable for the horizontal axis and one for the vertical, is variously called a *scatterplot*, a *scatter diagram*, a *scatter graph*, a *2-D dotplot*, or a *star field*. As visualizations become more complex, the problem becomes worse, with no accepted standard names. In fact, the tendency has been in the field to come up with rather idiosyncratic names – perhaps so that trademarking them is easier. This, however, puts a large burden on newcomers to the field and does not help in understanding the differences and similarities between a variety of methods of displaying data.

---

<sup>1</sup>Results from this survey have been published in a number of articles and several books, of which the reference cited above is only one of many interesting articles.

There have been a number of attempts to form taxonomies, or categorizations, of visualizations. Most software packages for creating graphics, such as Microsoft Excel™, focus on the type of graphical element used to display the data and then subclassify from that. This has one immediate problem in that plots with multiple elements are hard to classify (should we classify a chart with a bar and points as a bar chart with point additions, or instead classify it as a point chart with bars added?). Other authors such as Shneiderman [98] have started with the dimensionality of the data (1-D, 2-D, etc.) and used that as a basic classification criterion. Recognizing the weakness of this method for complex data, Shneiderman augments the categorization with structural categorizations such as being treelike or a network. This lack of orthogonality makes it hard to categorize a 2-D network or a 3-D tree – which one is the base classification? Again we are stuck in a false dichotomy – a 3-D network view is both 3-D and network, so such a classification system fails for that example.

Visualizations are too numerous, too diverse, and too *exciting* to fit neatly within a taxonomy that divides and subdivides. In contrast to the evolution of animals and plants, which did occur essentially in a treelike manner, with branches splitting and subsplitting, information visualization techniques have been invented more by a *compositional* approach. We take a polar coordinate system, combine it with bars, and achieve a Rose diagram [82]. We put a network in 3-D, or apply a projection to an  $N$ -dimensional point cloud to render it in two dimensions. We add color, shape, and size mappings to all the above. This is why a traditional taxonomy of information visualization is doomed to be unsatisfying. It is based on a false analogy with biology and denies the basic process by which visualizations have been created: composition.

For this reason this book will follow a different approach. We will consider information visualization as a *language* in which we compose “parts of speech” into sentences of a language. This is the approach taken by Wilkinson in *The Grammar of Graphics* [134]. Wilkinson’s approach can most clearly be seen by analogy to natural language grammars. A *sentence* is defined by a number of elements that are connected together using simple rules. A well-formed sentence has a certain structure, but within that structure, you are free to use a wide variety of nouns, verbs, adjectives, and the like. In the same way, a *visualization* can be defined by a collection of “parts of graphical speech,” so a well-formed visualization will have a structure, but within that structure you are free to substitute a variety of different items for each part of speech. In a language, we can make nonsensical sentences that are well formed, like “The tasty age whistles a pink.” In the same way, under graphical grammar, we can define visualizations that are well formed but also nonsensical. With great power comes great responsibility.<sup>2</sup>

---

<sup>2</sup>One reason not to ban such seeming nonsense is that you never know how language is going to change to make something meaningful. A chart that a designer might see no use for today becomes valuable in a unique situation, or for some particular data. “The tasty age whistles a pink” might be meaningless, but “the sweet young thing sings the blues” is a useful statement.

In this book, we will not cover grammar fully. The reader is referred to [134] for full details. Instead we will simply use grammar to let us talk more clearly about visualizations. In general, we will use the same terms as those used in grammar, with the same meaning, but we will omit much of the detail given in Wilkinson's work. Here we will consider a visualization as consisting of the following parts:

**Data** The data columns/fields/variables that are to be used

**Coordinates** The frame into which data will be displayed, together with any transformations of the coordinate systems

**Elements** The graphic objects used to represent data; points, line, areas, etc.

**Statistics** Mathematical and statistical functions used to modify the data as they are drawn into the coordinate frame

**Aesthetics** Mappings from data to graphical attributes like color, shape, size, etc.

**Faceting** Dividing up a graphic into multiple smaller graphics, also known as paneling, trellis, etc.

**Guides** Axes, legends, and other items that annotate the main graphic

**Interactivity** Methods for allowing users to interact with the graphics; drill-down, zooming, tooltips, etc.

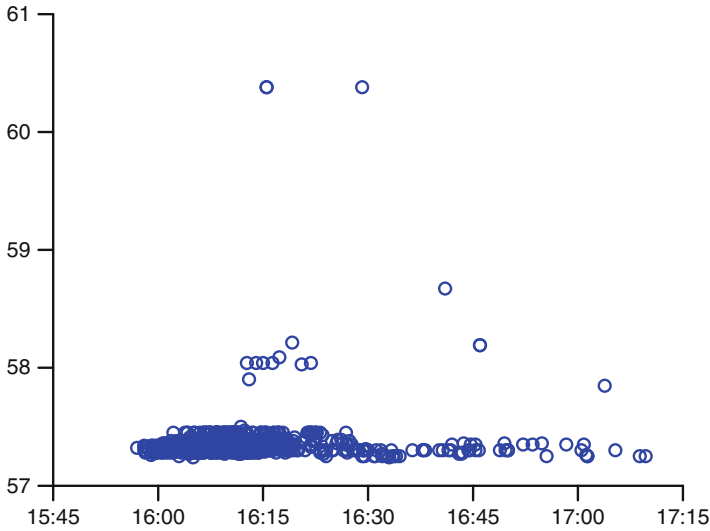
**Styles** Decorations for the graphic that do not affect its basic structure but modify the final appearance; fonts, default colors, padding and margins, etc.

In this language, a scatterplot consists of two variables placed in a 2-D rectangular coordinate system with *axes* as guides and represented by a *point* element. A bar chart of counts consists of a single variable representing categories, placed in a 2-D rectangular coordinate system with *axes* as guides and represented by an *interval* element with a *count* statistic.

Because the grammar allows us to compose parts in a mostly orthogonal manner, one important way we can make a modification to a visualization is by modifying one of the parts of the grammar and seeing how it changes the presentation of the data. In the remainder of this chapter, we will show how the different parts can be used for different purposes, and so introduce the terms we will use throughout the book by example while providing a brief guide to their use.

## 2.2 Elements

In a traditional taxonomy as presented by most computer packages, the *element* is the first choice. Although we do not consider it as quite that preeminent, it makes a good place to start with our exploration of how varying the parts of a visualization can change the information it provides and thus make it easier or harder to understand and act on different patterns within the data.

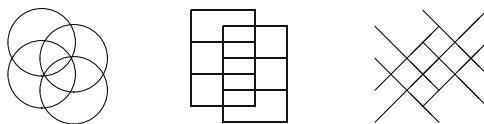


**Fig. 2.1** Stock trades: price by time. A scatterplot: two variables in a 2-D *coordinate* system with axes; each row of the data is represented by a *point*. The data form a subset of trade data for a single stock, with each point representing the time of a trade and the price at which it was traded.

### 2.2.1 Point

The *point* element is the most basic of elements. A single, usually simple, mark represents a single item. In the earliest writings, tallies were used for counting, with a one-to-one mapping between items and graphical representation. This basic representation is still a valuable one. Figure 2.1 shows a scatterplot depicting stock trades. Each point indicates a trade, with the  $x$  dimension giving the time of the sale and the  $y$  dimension the price at which the stock was traded. Some things to notice about this figure:

- Using points, all the trades are individually drawn. This has the advantage that you can see every item. This means that the times where there are many trades are easily visible. However, it has the disadvantage that quite a few points are drawn on top of each other, making a dense region where it is hard to see what is going on. This is often called the *occlusion problem*.
- The symbol used to draw the point makes quite a difference. Here we have used an *unfilled circle*. This is generally a good choice, especially for dense plots like this one. Overlapping circles are much easier to distinguish than symbols with straight edges – the eye can easily distinguish two, three, or even four overlapping circles. However, the same number of overlapping squares or crosses is confusing:



- The size of the points makes a difference. A good guideline is that the size of the points should be about 2 or 3% of the width of the frame in which the data are being drawn, but if that makes the points too small, it may be necessary to increase that size somewhat. If there are few points to be drawn, a larger size can be used if desired.

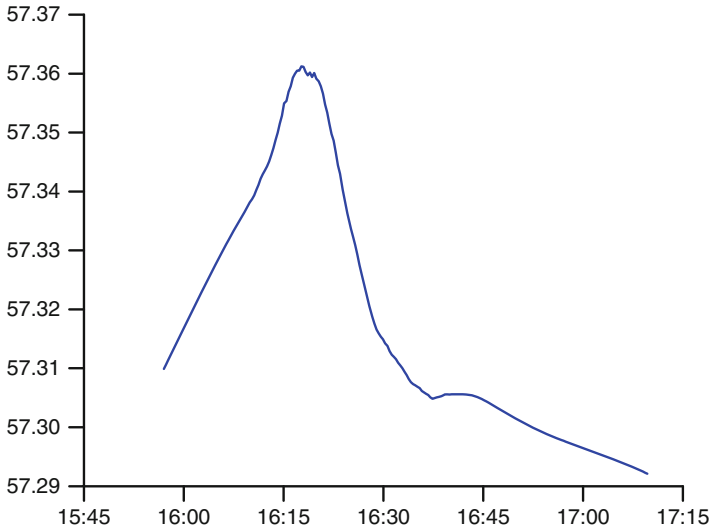
### 2.2.2 Line

*Lines* are a fundamentally different form of graphical element from points. When we use a point element, each case or row of data is represented by a single, discrete graphical item. For a line element, we have a single graphical element that represents many different rows of data. From a theoretical point of view, a line represents a function:  $y = f(x)$ . In other words, each value of  $x$  can have only a single value of  $y$ . This has several important ramifications:

**Lines usually require a summary statistic.** Because a line must have a unique  $y$  value for each  $x$  value, some method of aggregation or summarization is required to use a line element on data with multiple values for the same  $x$  location. Compare Fig. 2.1 with Fig. 2.2. Especially between 4:00 and 4:30 there are many points with the same  $x$  value. To allow the line element to represent a single value for each  $x$  value, we have applied a statistic to the data to create a summary value for each  $x$  location. In this figure we have used a loess smoother to smooth the data.

**Lines interpolate between values.** A line is defined over a continuous range of values, but data typically consist of a finite set of values, so between recorded values of  $x$  a line interpolates data. In Fig. 2.2 the interpolation is explicit in that there is a smooth statistic applied, but even in a simple line chart where the data have only single rows for each value of  $x$ , and so a statistic is not required, interpolation is necessary. Drawing a line between  $x$  values makes at least the implicit assumption that such an interpolation makes sense; if the stock value at 5:00 is 57.30 and the value at 5:02 is 57.29, then using a line element only makes sense if it is reasonable to assume that the stock value at 5:01 was both defined and somewhere reasonably close to the range [57.29, 57.30].

The last point above has a corollary: Lines are generally not an appropriate representation for categorical data. If the  $y$  values are categorical, then a simple line element gives the impression that as  $x$  changes, the quantity being plotted smoothly changes between different categories, which is not possible. This impression can simply be accepted as necessary for a given representation, or an interpolation



**Fig. 2.2** Stock trades: price by time. Line chart: two variables in a 2-D coordinate system with axes; a single *line* represents all the data. A smooth statistic (Sect. 2.3) has been applied to the data. The data are the same trade data of the previous figure

method can be used that shows abrupt changes, such as a “step” style drawing, as given in Fig. 2.3.

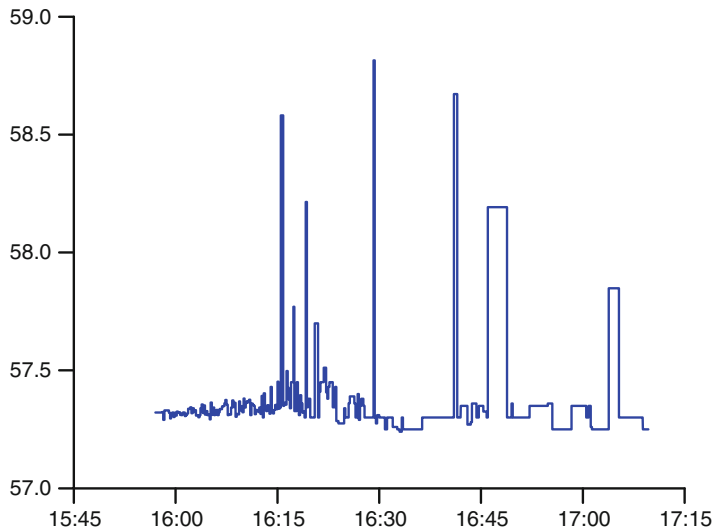
If the  $x$  values are categorical, the situation is worse. By its nature the line element must interpolate along the  $x$  dimension, so the chart will be intrinsically misleading. It is important to note that in many cases data that may appear categorical are based on an underlying dimension that is continuous. Values of time, for example, might be recorded as a set

$$\{\text{Monday}, \text{Tuesday}, \text{Wednesday}, \text{Thursday}, \text{Friday}, \text{Saturday}\},$$

which are categories. These categories, though, represent ranges of time – an underlying dimension that is continuous. Therefore, a line element using the values given above on the  $x$  dimension is a reasonable chart. It is only when the underlying dimension cannot be thought of as continuous that the result loses meaning.

### 2.2.3 Area

An *area* element is most simply defined as filling the area between a line and the horizontal axes. The simplest area element is indeed just the area under a line element, and if we replaced the line element in Fig. 2.2 with an area element, the

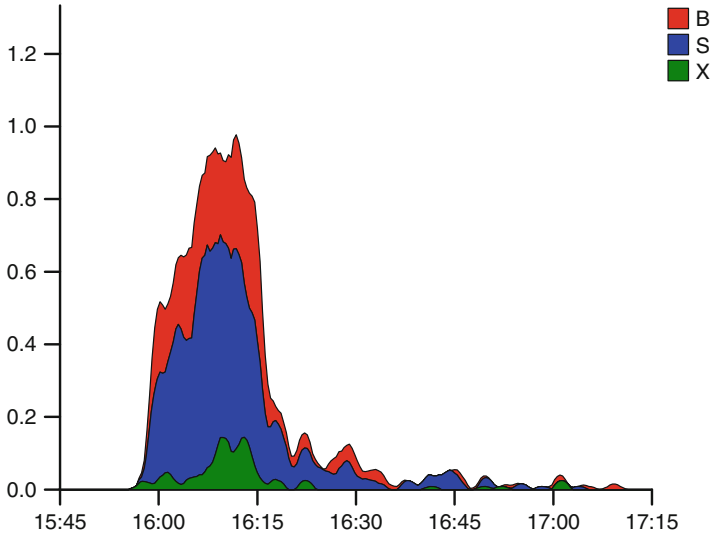


**Fig. 2.3** Stock trades: price by time. Step representation of a line chart. This is the same chart as in Fig. 2.2, except that we have used a step function on the data so it does not interpolate smoothly between values, but instead steps abruptly

chart would be essentially the same as if we filled in below the curve using a paint tool in a graphic editing program.

Given their similarity, the question needs to be asked: Is there any real difference between the two elements, or can we treat them the same? When there is a single line or area in a chart, there is indeed little reason to prefer one over the other, but when there are multiple lines or areas – for example, when an aesthetic (which we will look at in Sect. 2.4) splits the single line or area into several – there is a difference, as follows.

- Areas are more suitable than lines when the  $y$  value can be summed, for example, when the  $y$  values represent sums, counts, percentages, fractions, density estimates, or the like. In these situations, areas can be stacked, as in Fig. 2.4. This representation works well when the overall value is as important as, or more important than, the relative frequencies of the  $y$  values over time. If the relative frequencies are of greater interest, instead of showing a summation of  $y$  values, we can show relative proportions as in Fig. 2.5.
- Lines are more suitable for areas when the  $y$  values should not be summed, or when there is a need to compare the values for different lines to each other, or to compare their shapes. Areas that are not stacked tend to obscure each other and so are unsuitable for such uses.
- Areas can be defined with both lower and upper bounds, rather than having the lower bound be the axis. This representation is particularly suitable for representing ranges that vary along the  $x$  dimension, such as is often the case for

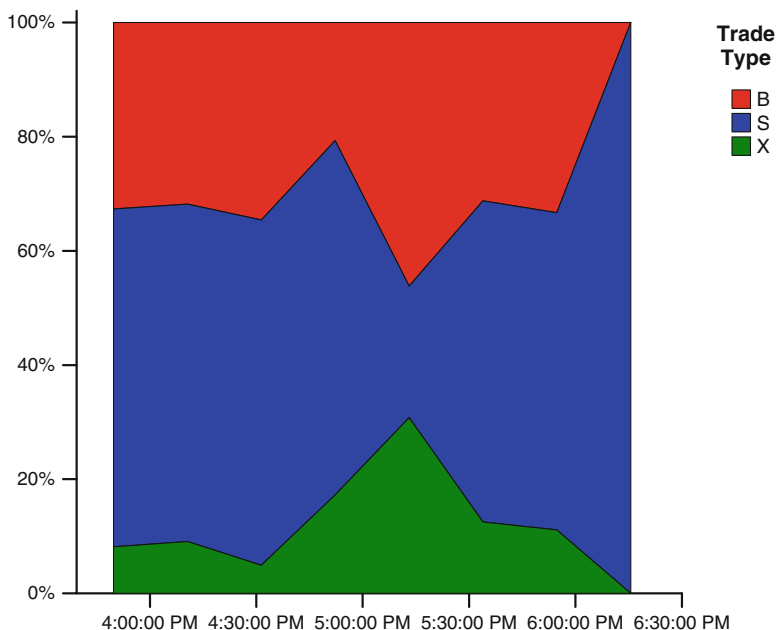


**Fig. 2.4** Stock trades: volume by time. An area chart: two variables in a 2-D coordinate system with axes; an area element is displayed for each group in the data. The groups are defined by the *TradeType* variable, which indicates whether the trade was a buy, sell, or cross-trade. For each group, an area element represents the relative density of trades over time. The areas are stacked on top of each other, so the top of the stacked areas gives the overall density of trades over time, while the bands give the relative numbers by type. Note that in this chart it is relatively easy to visually estimate the total height of the stacked element, and also to see the shape of the lowest band, because it is anchored to the line. It is the other categories, *buy* and *sell*, that are hard to judge as their baselines are stacked on other areas

quality control charts, and for representing statistical ranges such as deviations about a model fit line.

- Consideration should also be paid to the variable being plotted on the  $x$  and  $y$  axes. The “area” of the area element should have some sort of meaning. In other words, consider the units of the 2-D area. If it has some reasonable meaning, then an area element makes sense. Otherwise, it might be best not to use an area element. For example, if the  $x$  dimension is *time*, and *velocity* is on the  $y$  axis, then the area of an area element has a direct interpretation as  $velocity \times time$ , which is distance traveled, making the chart reasonable. On the other hand, an area chart of  $starttime \times endtime$  would be a bad choice as the area is meaningless.
- If the concern is to see how a value is changing over time, then using a line is often a better choice, as the slope of the line is the rate of change of the  $y$  variable with respect to the  $x$  variable. If acceleration is of greater interest than distance traveled, then a line element is a better choice than an area element in the same situation as discussed just above, where  $x = time$  and  $y = velocity$ .





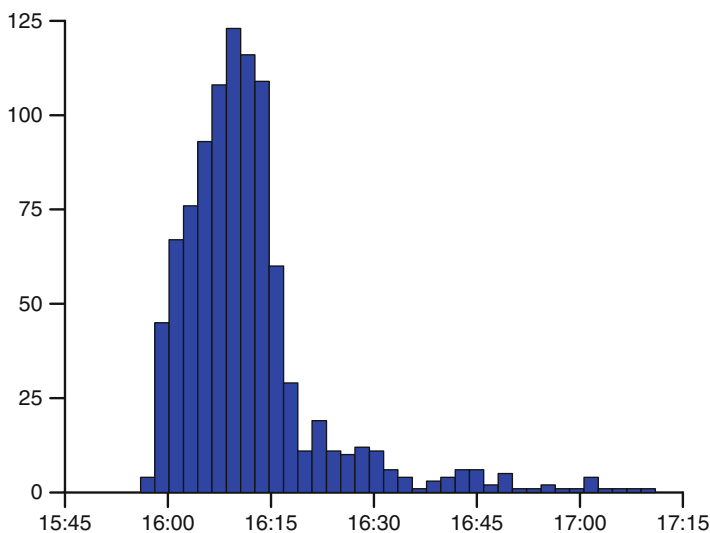
**Fig. 2.5** Stock trades: ratios of types by time. A modified version of Fig. 2.4 in which the density statistic has been replaced by a statistic that first bins the data using the horizontal (time) dimensions and then calculates the percentage of each group within each bin. The result shows the changing proportions of trades that were buys, sells, or cross-trades

### 2.2.4 Interval

Intervals are typically termed *bars* when in a rectangular coordinate system and can be used in a variety of ways. They can be used, like points, with one bar to every row in the data set, but that use is relatively rare. Often they are used to provide a *conditional aggregation* where we aggregate a set of rows that share the same  $x$  dimension. The canonical example of this use of an interval is the “bar chart,” where a categorical variable is used on the  $x$  axis, and where the  $y$  values for each distinct  $x$  axis category are summed, or, if there is no  $y$  value, the count of rows in each category is used.

One special case of the “bar chart” is when we have a continuous variable on the  $x$  dimension and wish to show a visualization of how concentrated the occurrences are at different locations along that dimension. We bin the  $x$  values and then count the number of values in each bin to form a  $y$  dimension. The common name for this chart is a *histogram*, as shown in Fig. 2.6.

Compare Figs. 2.6 and 2.4. Their overall shape is similar – we could easily add a color aesthetic to the histogram to obtain a plot that has the same basic look as the density area chart. This illustrates not only the fact that the histogram is a form



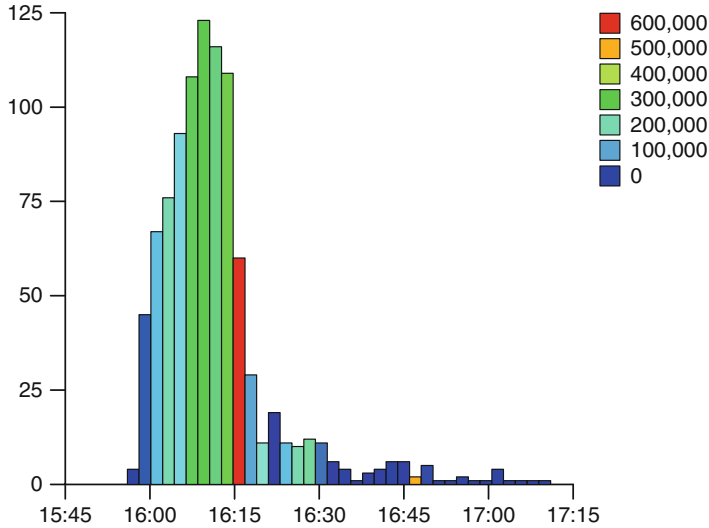
**Fig. 2.6** Histogram of trade times: one data variable in a 2-D coordinate system. The second dimension is generated from the first by a pair of statistics. The first statistic bins the  $x$  values into disjoint bins, and the second statistic counts the number of rows that fall in each bin. This gives a histogram representation of when trades occurred. The data form a subset of trade data for a single stock, with each point representing the time of a trade and the price at which it was traded.

of density statistic but also the similarities between the area and bar elements. In many respects, a bar can be considered as half-way between a point and an area element, sharing the abilities of both. Perhaps this is why it is the most commonly used element in published charts. The main reason to prefer an area element over an interval element is for accentuating the continuous nature of the  $x$  dimension. The interval element breaks up a continuous  $x$  dimension into chunks, ruthlessly splitting the data into (often arbitrary) categories, whereas the area element renders a single, smoothly evolving value. On the other hand, if you want to show additional information on the chart, the bars are more versatile and will allow you to make more complex visualizations.

Figure 2.7 takes the basic histogram and adds some more information. We have used a *color aesthetic* and used color to show the volume of trades in each binned time interval. In this visualization we show the count of trades as the main focus and relegate the trade volume to secondary status as an aesthetic. In practice the converse is likely to be a better idea.<sup>3</sup>

Many published books (e.g., [38]) and, increasingly, Web articles will tell their readers to always show the zero value on the  $y$ -dimension axis when drawing

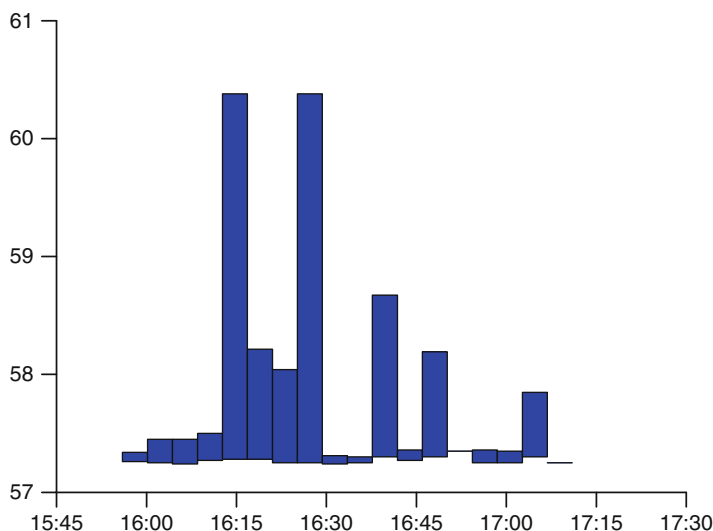
<sup>3</sup>In Sect. 2.3 we will explain a little more about statistics. In particular we will deal with the use of weight variables, the use of which is the best way to describe this data set for most purposes.



**Fig. 2.7** Histogram of trade times. This figure is similar to Fig. 2.6, but we have colored each *bar* by the sum of trade volumes in each *bar*. We can see in this figure that, although most trades took place between 4:10 p.m. and 4:15 p.m. (local time) the time period just after this period saw more total trade volume

a bar chart. While the advice is not bad advice when stated as a general guideline, be aware that it is not a universal rule – there are important exceptions. The rule is based on the principle that the length of the bar should be proportional to the quantity it measures, and so an axis value that is not at zero misleads by showing only parts of those bars, exaggerating differences. Instead, consider if zero really is meaningful for your data and if it is important to be able to compare lengths. In most cases, the answer is yes and the advice is good, but, like all advice, do not follow it slavishly.

**Zero may not be a good baseline.** Consider a chart where  $x$  represents buildings in Chicago and  $y$  the altitude of their roofs as measured above sea level (a subject of some interest to the author as he types this on a windy day in the Sears Tower). A more natural baseline than zero would be the average altitude of the city itself, so the heights of the bars would more closely approximate the heights of the buildings themselves. Other examples of  $y$  dimensions for which zero is not necessarily a natural base point are temperatures (in Celsius and Fahrenheit), clothing sizes, and distances from an arbitrary point. Often falling into this case are charts where the  $y$  dimension has been transformed. For a traditional log scale, for example, it is impossible to show zero, and showing the transformed value zero (the original value “1”) is as arbitrary a baseline choice as showing some other location and might be completely inappropriate if you have data values below one.



**Fig. 2.8** Range plot of trade times: two variables in a 2-D coordinate system with two chained statistics. The first statistic bins the  $x$  values into disjoint bins and the second statistic calculates the range of  $y$  values in each bin. This gives a representation of the range of trade prices over time. The data form a subset of trade data for a single stock, with each point representing the time of a trade and the price at which it was traded.

**Differences are more important than absolute quantities.** If you are preparing a visualization in which your goal is to highlight differences, and the absolute values of the quantity are of little interest, then it makes sense to focus on the range of differences rather than showing a set of bars all of which appear to be about the same size. If you are tracking a machine that is expected to make between 980 and 1020 items per minute, a bar chart with a zero on the  $y$  axis would make a much weaker tool for quality control than one that shows a range of [950, 1050].

**The intervals represent a range, not a quantity.** Many statistics produce an interval that is not fixed at zero. Examples are ranges, deviations, and error bars. Because these intervals represent a spread around a mean or, more generally, around a central point of the  $y$  data for a given  $x$  value, they should be thought of more as a collection of those central points, and zero is unlikely to be an important part of their range.

Figure 2.8 illustrates the last two points. The bars represent a range of  $y$  values, rather than a single quantity, so we should consider the underlying quantity – the trade price itself. Now zero is indeed a natural baseline for prices, so it would be defensible to use it as the  $y$ -axis minimum value. However, in stock trading (at least for intraday trading) differences are much more important than absolute values, so a range that shows the differences is preferable to a range that hides them by showing the absolute values. For these data and this statistic, zero is a poor choice.

Looking at the plot of ranges of stock prices, we can see that they are quite large for some time periods. Is that because of a few outlying trades, or was it a short-lived trend? What we want is some way of drilling deeper into those bars and drawing out the distribution inside each one.

### 2.2.5 Schema

One tool for summarizing a distribution in this way was invented in the mid 1970s and popularized by John Tukey [112] – the boxplot. The *boxplot*, also known as the “box and whiskers plot,” is an example of a *schema*. A schema is a graphic element that produces an iconlike representation of data. In a boxplot, a box is drawn enclosing the middle 50% of the data, with a line drawn inside it at the median. The “whiskers” are drawn outside this box so as to enclose all data that might be considered normal. Data outside the whiskers are classified as outliers and drawn as points in two styles to represent “regular” and “extreme” outlying data points.<sup>4</sup>

Figure 2.9 shows the same information as Fig. 2.8, but with the interval element replaced by a boxplot schema element. The relationship between the two elements should be clear, and we can see that the range was indeed due to some high-priced outliers. If we zoom in on the y dimension to exclude these points (Fig. 2.10), we see that, apart from some extreme trades, the price has remained relatively stable.

This plot highlights one of the strengths of the boxplot – the use of *robust statistics* like the mean and interquartile range (the middle 50% of the data). The boxplot allows us to see the trend undistorted by the outliers, but also allows us to see those same outliers.

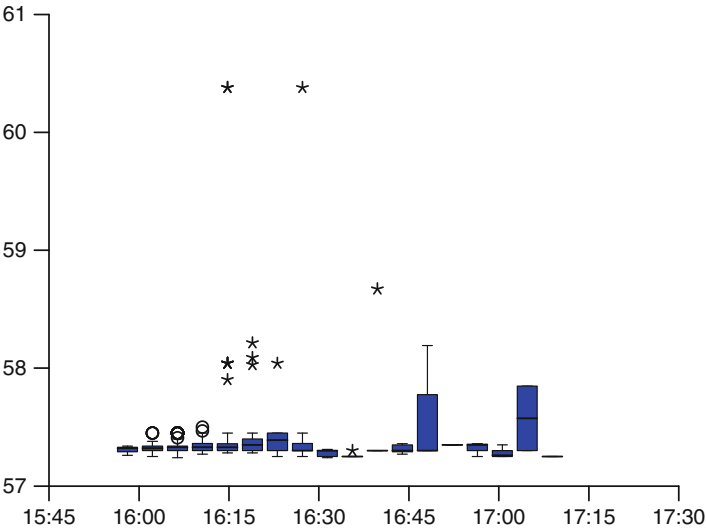
Other types of schema, such as Chernoff faces [20], are of less use for visualizing time series data, being more appropriate to specialized systems with known data. It would be possible to use high-dimensional glyphs (such as Chernoff faces) for high-dimensional time series, but so far few compelling examples of such use have been demonstrated.

### 2.2.6 Multiple Elements

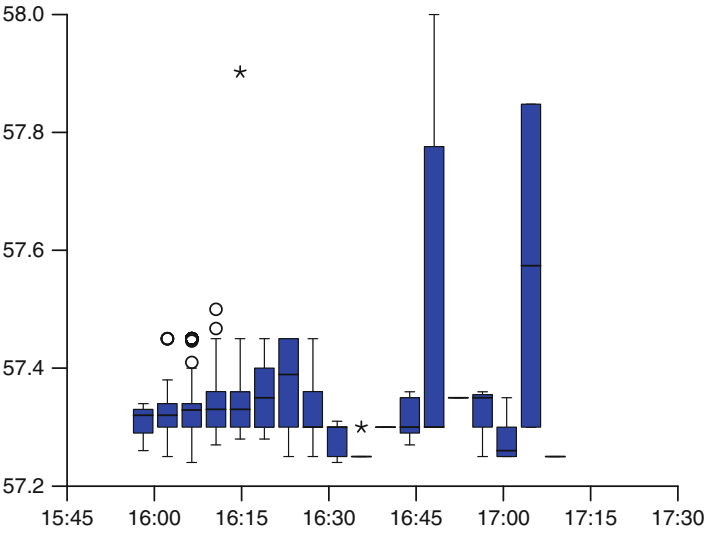
Figure 2.11 shows a final important point concerning elements. Combining multiple elements into the same chart can be an effective way to display different aspects of

---

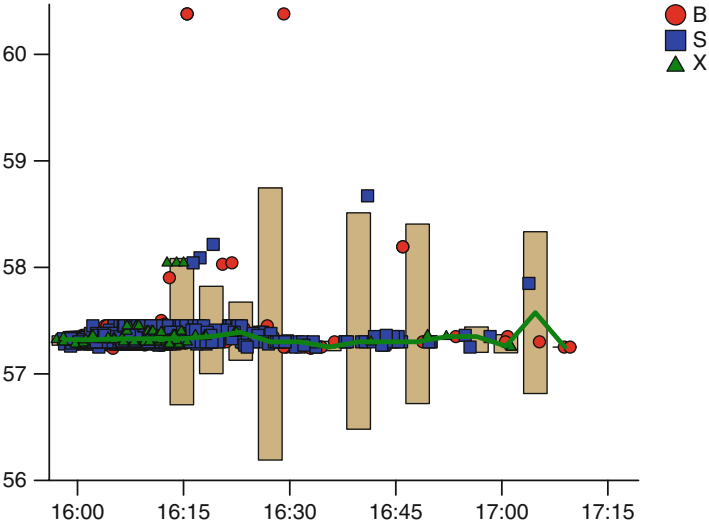
<sup>4</sup>The details of drawing a boxplot are technically quite tricky, especially in the presence of weighted data, for which Tukey does not provide much help. For large amounts of unweighted data these details may not be apparent, but for small data sets and for weighted data sets it is possible to get different displays from different graphical packages. However, since the boxplot was designed primarily for exploring data, these minor technical differences should not affect the overall goal of discovering patterns and trends.



**Fig. 2.9** Stock trades: price by time. Boxplot: two variables in a 2-D coordinate system with two chained statistics. The first statistic bins the  $x$  values into disjoint bins and the second statistic calculates the *Tukey statistics* of  $y$  values in each bin. The data form a subset of trade data for a single stock, with each point representing the time of a trade and the price at which it was traded.



**Fig. 2.10** Boxplot: the same graph as in Fig. 2.9, but restricting the  $y$  dimension to show a smaller range.



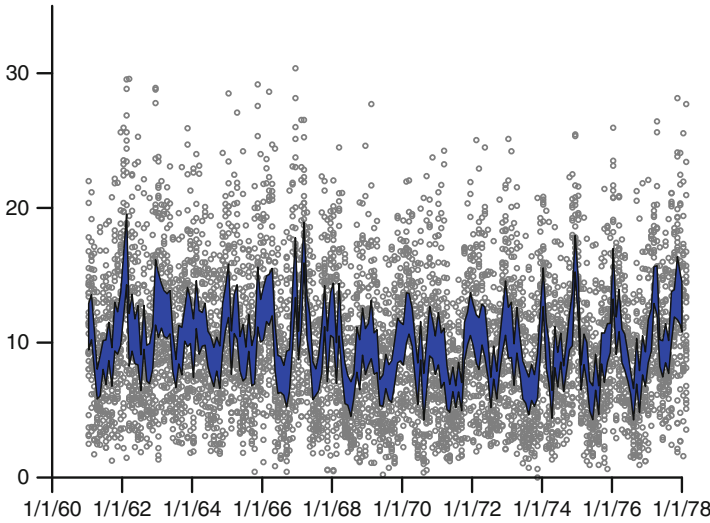
**Fig. 2.11** Combination of three elements displaying trade price by time in two dimensions. A point element showing individual sales, an interval element showing, for binned values, the 95% confidence interval for those values, and a line element showing the median values in the same bins

data simultaneously. With three different elements, Fig. 2.11 requires some study but does provide a view of the central trend of the price, as well as estimates of variability and the finest level of detail possible; points show the individual trades.

Combinations of elements is usually permitted to some extent with traditional, chart-type-based graphing packages, but it is restricted to the most common examples, such as bar/line, bar/points, and point/line. Allowing a freer mixture of elements allows increased creativity and permits visualizations more easily tailored to the goals of the data, but even with a limited set of choices element combination allows multiple properties of data to be shown within a single chart. By carefully allocating variables to elements, and by choosing suitable statistics for those elements, compelling and informative visualizations can be produced.

2.3 Statistics

In Sect. 2.2 we saw the use of statistics in the context of choice of element. We define a *statistic* for a visualization very generally as any transformation or chain of transformations that take data and produce new forms of it. In Fig. 2.12 we see a simple example; we took a data set consisting of wind speed measurements over 18 years, divided it into months, and then calculated a 95% confidence interval for



**Fig. 2.12** Wind speeds measured daily in Dublin, Ireland for the years 1961–1978. On *top* of the raw data is superimposed an area element showing a 95% confidence interval for the mean wind speed in each month

the mean of the data for each month. The resulting statistics are shown using an area element to highlight the continuously changing nature of the statistic being calculated.

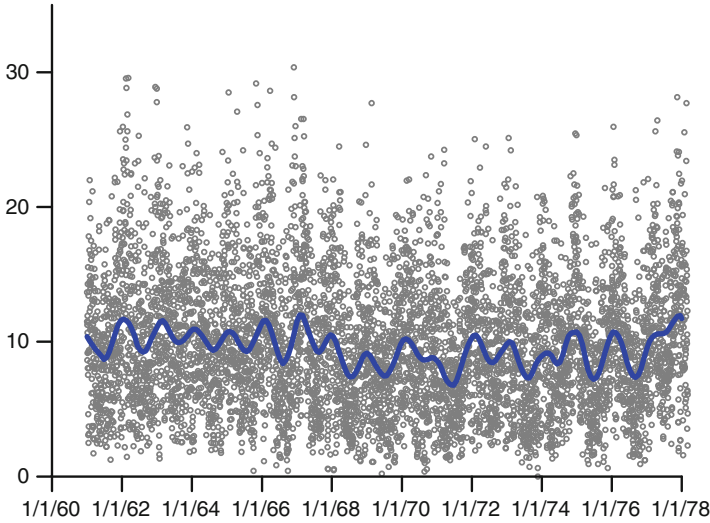
This is a fairly traditional usage of statistics; the statistic summarizes the data in a given group by a pair of values (indicating the range). We are summarizing the 28 to 31 points in each month by a single range. This is how statistics are normally thought of – as a means of summarizing information. However, in this book we also consider more general uses, as discussed in the subsections below.

### 2.3.1 Local Smooths

A common way of summarizing a set of  $(x, y)$  data where we expect  $y$  to depend on  $x$  is to fit a line to the relationship. The traditional family of prediction lines to fit are *polynomial least-squares-fit lines*. These summarize the relationship between the variables using a polynomial of low degree. However, for time data in particular this form of smooth<sup>5</sup> is unlikely to be useful. When  $x$  represents time, it is not common to have a function linearly increasing or decreasing over time. Using a

<sup>5</sup>In this context, the terms *smooth* and *predictor* are being used interchangeably. To be more formal, we would say that a smoothing statistic *can be used* as a predictor, rather than using the two terms interchangeably.





**Fig. 2.13** Wind speeds measured daily in Dublin, Ireland for the years 1961–1978. The figure shows both the raw data and a loess smooth of the data. The loess smooth uses a window size of 1 year and an Epanechnikov kernel

higher-degree polynomial does not improve the situation much. Often a fit will be needed for a seasonal variance or other nonlinear structures. But even these models will fail if the data have change points (where the relationship changes abruptly) and the calculation of seasonal models is itself a challenge. Often what is needed is a way of smoothing the data that can be used to take an exploratory look at the series. A desirable feature of such smooths is that they adapt to different conditions in different time periods, unlike polynomial fits, which expect the same conditions to hold across the entire range of time. *Local smooths* are statistics that only use data close to the  $x$  value where the smooth is being evaluated to calculate the smoothed  $y$  value. One very simple smooth is a *moving average*. For each  $x$  value we produce a  $y$  value by averaging the  $y$  values of data lying within a given distance of the  $x$  value on the  $x$  dimension.

Figure 2.13 shows a *loess smooth* applied to data. Loess [26] adapts basic regression by calculating a new regression fit for each value of  $x$  by fitting a regression line only to the data within a given distance of that value, and with decreasing weights the further away we get from that central location. For this example, the distance used is fixed at 1 year, so when we predict a value at 1972, we only calculate the fit for data within 1 year of that date, and we weight values closer to 1972 higher than values further away. We see the seasonal variation clearly in this figure; a loess smooth is a good tool for exploratory views of time series, although the results are heavily dependent on the choice of the given distance. There are many options for choosing this “given distance,” usually termed a *window*, including:

**Fixed- or variable-sized window:** The simplest window is of fixed size; we can state that the window should be 1 month long, 1 week long, or some fraction of the data range. Alternatively we could ask for a variable or adaptive window. One way of doing that is by defining a *nearest-neighbor window* in which we define the local width at a point on the  $x$  dimension as the distance necessary to enclose a fixed number of neighbors. If the data are irregularly spaced on the  $x$  dimension, this allows the window to adapt to the relative density, so that sharp changes are possible in high-density regions, but low-density regions, for which less information is available, do not exhibit such variability.

**Window centered or not:** Generally, if we want to predict at a given location, we choose a window of values centered on the value we want to predict. For time data, however, this makes less sense as it is trying to predict a value based on values that are “in the future.” A more advisable plan is to use a window that only calculates values to the left of the location where we want to predict. This represents reality better, and that is the basic goal of modeling.

**Choice of kernel function:** The kernel function is a function that gives more weight to observations close to the location and less to ones far away. The simplest kernel function is a uniform kernel, which gives equal weight throughout the window. A triangle kernel function that decreases linearly to zero as observations get further away from the current location is another simple one. In general, the choice of kernel function is not as important, especially for exploratory use, as other choices, so pick a reasonable choice and then forget about it. In this book we use the *Epanechnikov* kernel throughout.<sup>6</sup>

Without doubt, the most important choice to be made when using a kernel smooth is the window size. In Fig. 2.13, we can see seasonal trends; the window size of 1 year allows us to see fluctuations at this level of detail. In contrast, Fig. 2.14 has a larger window width and the seasonal details are hidden, showing only a simpler message of “little long-term trend change.” Both messages are valuable; deciding which to display is a matter of choosing the one that highlights the feature you are interested in conveying.

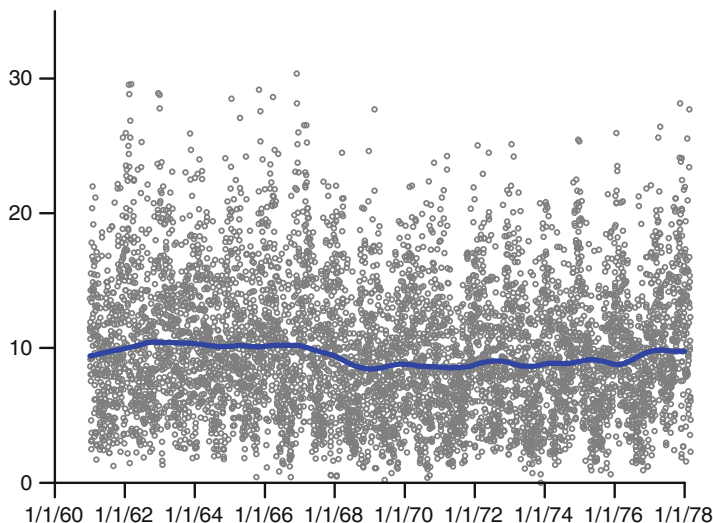
There is a large body of literature on how to choose window widths, paralleled by a similarly large body of literature on how to set binning sizes for histograms, which is a related topic. Although the initial choice of parameter is important in a production setting where you have set values and is very important when you want to compare different data sets or make inferences, in a more exploratory setting the

---

<sup>6</sup>The formula for an Epanechnikov kernel is defined as 0 outside the window and

$$\frac{3}{4} \left( 1 - \left( \frac{x}{h} \right)^2 \right)$$

when  $-h < x < h$  for a window width of  $h$ . This kernel minimizes the asymptotic mean integrated squared error and can therefore be thought of as optimal in a statistical sense.



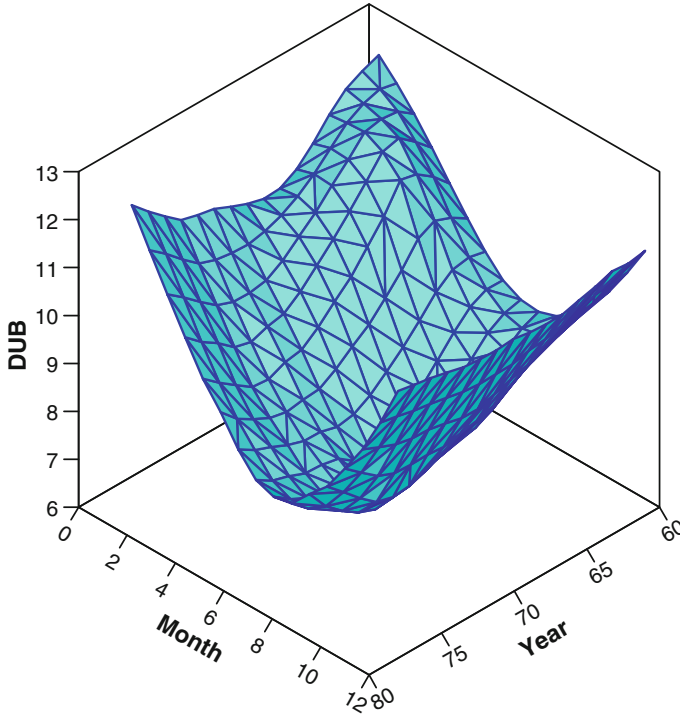
**Fig. 2.14** Wind speeds measured daily in Dublin, Ireland for the years 1961–1978. The figure is the same as Fig. 2.13, except that the loess smooth uses a window size of 3 years, which smooths out the seasonal trends

best solution is to start with a good value and then play around with the value to see what effect it has. A system that allows direct interaction with the window width parameter, as we will see in Chap. 9, is ideal for such exploration.

In the context of the grammar, we want our choice of statistic to be orthogonal to our other choices; in particular, this means we would like to be able to use our statistic in any dimensional space. Our conditional summary statistics should be conditioned on all the independent dimensions, and if we have a smoothing statistic, we should ensure it works in several dimensions. Figure 2.15 shows our loess smooth in three dimensions. By taking our 2-D *date*  $\times$  *value* chart and splitting the dates into two components, one for months and one for years, we highlight the seasonality. We will give more examples of this technique in Chap. 5.

### 2.3.2 Complex Statistics

I do not want to give the impression that all statistics are simple summaries or smooths and predictions that essentially enhance the raw data. In this book statistics are considered to be far more general objects that can make radical changes to the structure and appearance of graphics. In some systems, such statistics are not available inside the graphical framework and are performed before any other part of



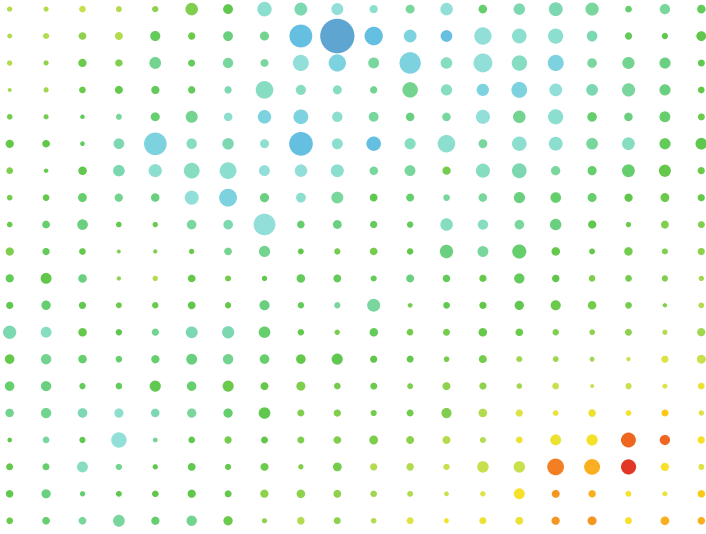
**Fig. 2.15** Wind speeds measured daily in Dublin, Ireland, by year and month. The *date* variable has been split into two dimensions, one for year and one for month. A surface for the loess smooth has been plotted

the process, but the principle is still applicable. Figure 2.16 shows an example of a more complex statistic, the *self-organizing map* (SOM) invented by Teuvo Kohonen. The SOM is described in detail in his book [68], with what follows being a brief introductory description only.

A SOM first must be trained on data. This is a process that creates a grid of vectors that represent clusters for the data. The training algorithm for a simple SOM implementation on  $k$ -dimensional data table is as follows:

1. Create a grid of  $k$ -dimensional vectors and initialize their values randomly.
2. For each row in the data set, find the grid vector that is closest to that row where the distance is measured in the  $k$ -dimensional space of the columns of the table.
3. Modify the grid vector and any other grid vectors nearby by making their values more similar to the input row vector. Grid vectors closer to the target grid vector are changed more than ones further away.

The update steps 2 and 3 are repeated a number of times, each time with a decreasing weighting value so that the grid changes less. The end result is to create a *semantic map* in which the initially random vectors have been “trained” to be similar to the data vectors, and the vectors have simultaneously been laid out so that similar patterns tend to be close to each other. The resulting map can be used in many ways.



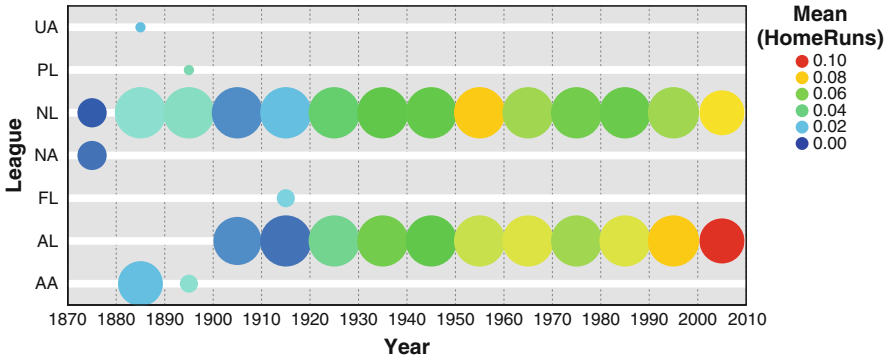
**Fig. 2.16** A self-organizing map of wind data using wind speed data for six cities as the six variables used to create the plot. The size of the *point* at a grid coordinate indicates how many rows are mapped to that location; the *color* represents the average wind speed for that grid location

In Fig. 2.16 we have taken the map and made one final pass on the data, assigning each data row to its closest grid location. Thus we have used the map to project our data from six dimensions down to a 2-D grid. Each grid point represents a number of days where the wind speeds measured at six cities in Ireland were similar. We summarize the data at each grid point with two values. The count of the number of days mapped to that grid point is shown as the size of the point; the average wind speed is shown by color, with red indicating high speeds, blue low speeds, and green intermediate.

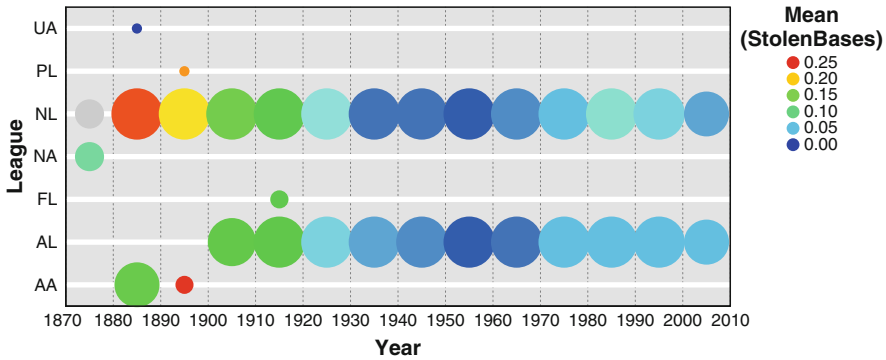
The resulting plot shows three main groups – semantic clusters. There is a relatively small group of days where there are high winds, but at least two groups where the average speeds are low. This gives some evidence that there might be a single weather pattern that leads to high winds, but multiple weather patterns on days when wind speeds are low. The SOM is not an exact tool; it is an exploratory tool that can be used to motivate more detailed study. It is also valuable in being able to reduce data dimensions and show a different aspect of time data, a subject we will return to in Chap. 8.

## 2.4 Aesthetics

In the last figure of the previous section, we used color to show average wind speed and size to show the number of days represented by each grid vector. These are examples of *aesthetics* – mappings from data to visual characteristics of the graphical elements.



**Fig. 2.17** Baseball players’ average number of home runs per game, binned into decades and split vertically by league. The color of the *points* encodes the average number of home runs per player per game



**Fig. 2.18** Baseball players’ average number of stolen bases per game, binned into decades and split vertically by league. The color of the *points* encodes the average number of bases stolen per player per game

Color is one of the most commonly used aesthetics; because visualizations are displayed visually (by definition!), it is always possible to color elements based on characteristics of the data. Color is not as dependent on the positions of the elements as other aesthetics such as shape, size, and rotation. Indeed, some elements (e.g., *area*) have much of their value destroyed when aesthetics such as rotation are applied, whereas color can be applied with ease.

In Figs. 2.17 and 2.18 we show color applied to baseball data. This data set consists of statistics on baseball players’ ability to hit the ball. The data are yearly sums of various measures of hitting ability, and we have restricted the data only to include players who played at least 50 games in a given season. This figure breaks

down players by *decade* and *league*, showing a point for each league/decade combination. The total number of player seasons for that league/decade combination is indicated by the size of the point.

We can see a flurry of smaller early leagues, settling down later into the two leagues now in existence; the American League and the National League. The first figure uses color to show the average number of *Home Runs* hit per game for a given league in a given decade. The second figure uses color to show average stolen bases.<sup>7</sup> We can see how home runs, rare initially, have become relatively more common in modern times, and we see a big increase in their occurrence since 2000 in the American League. Stolen bases, on the other hand, were much more common earlier, reached their nadir around 1950–1960, and since then have seen a slight, but clear, increase.

One feature of interest is the gray symbol for stolen bases in the National League in the 1870s. This is because there are no stolen base data for this element. It is important to make sure that people viewing a chart can clearly see missing data; we would not want to show this symbol using a color that might be confused with known data. When using a color aesthetic, achromatic colors (black, gray, white) are a good choice for missing values as they are easily distinguished and unlikely to give a misleading impression – unless you print the chart in black and white!

Much has been written about the uses of color in visualizations. Good starting points for deep investigation are [72] and [14]. The latter reference describes an excellent online tool for generating color scale, *ColorBrewer*, which can be found at [colorbrewer.org](http://colorbrewer.org). What follows are simple guidelines to keep in mind.

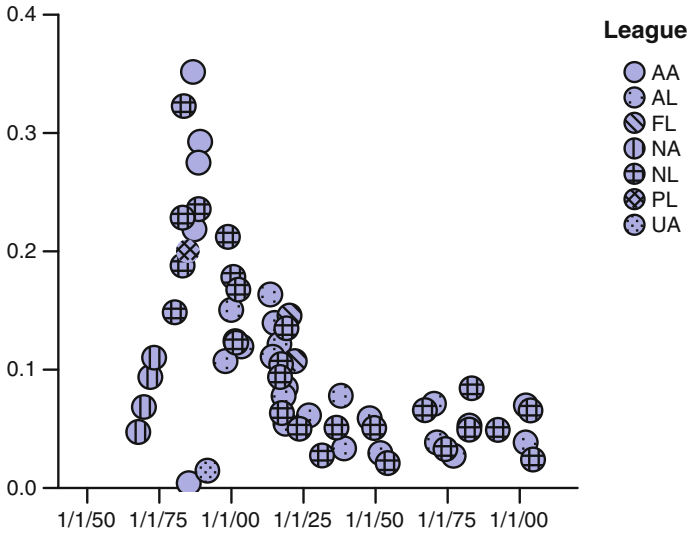
**Color is a composite.** A color can be thought of as being a mixture of red, green, and blue; cyan, magenta, yellow, and black; or as a point in more technical color spaces. It is tempting to think that we can use different components for different mappings. This can be done effectively, but it is tricky, as our visual tendency is to see colors as a single entity. Perhaps the most useful separation is to parameterize color by hue, saturation, and lightness (or a similar system with better perceptual properties, such as the CIE  $L^*U^*V^*$  space, also known as CIELUV), and use just a couple of the components. In Sect. 2.4.2 we show another example of a composite aesthetic.

**Color is nonlinear.** Mapping a variable to any simple path through color space will give a mapping where perceived differences in color are poorly related to actual differences in data. This makes it hard to make quantitative judgements about differences in values based on the displayed colors.

**My color is different from your color.** Not only might we have different abilities to perceive color differences (color-blindness in one form or another is an

---

<sup>7</sup>Home runs are big hits that in today are virtually always hit out of the field of play. Stolen bases, in contrast, are when the runner advances to the next base without making a hit at all. In a sense, they are opposite types of play that advance the bases for the batting team.



**Fig. 2.19** Baseball players' average number of stolen bases per game. This figure shows the same information as Fig. 2.18, namely, how many bases are stolen per player per game, aggregated into decades and split by league using a *pattern* aesthetic. The *points* have been jittered to reduce the effect of overlap

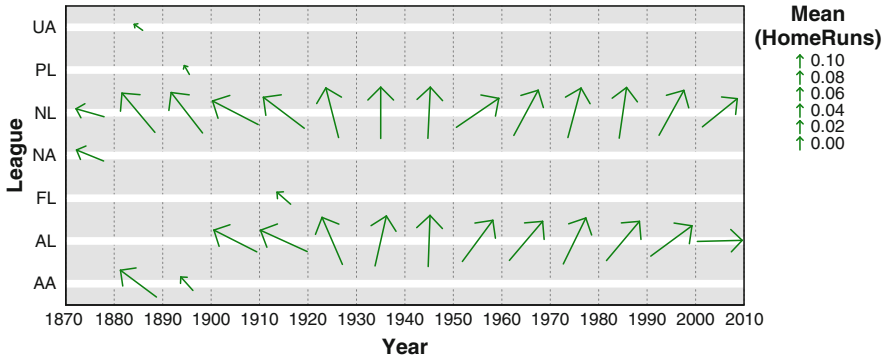
example of this), but my computer might have a different gamma from yours, my projector might show blues badly, and your printer might be greener than expected. Color calibration is big business and important for graphics professionals, but depending on accurate color for visualization results is a risky proposition. Err on the side of caution and do not expect your viewers to make inferences based on subtle shadings.

### 2.4.1 Categorical and Continuous Aesthetics

Figures 2.17 and 2.18 show a continuous value being mapped to color. In other figures, for example Fig. 2.5, we map data consisting of categories to color. In Chap. 4 we will consider the differences between displaying categorical and continuous values more thoroughly, but the difference between aesthetics for categories and aesthetics for continuous values is fundamental to the design of good visualizations and needs to be understood when talking about aesthetics.

Some aesthetics are naturally suited to categorical data. *Pattern*, also known as *texture*, is an examples of this. In Fig. 2.19 we have taken the same data as in Fig. 2.18 and moved variables to different roles. We have left time on the  $x$



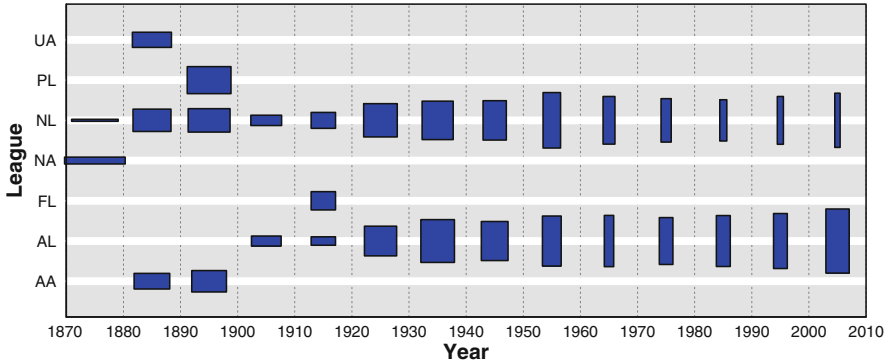


**Fig. 2.20** Baseball players’ average number of home runs per game, binned into decades and split vertically by league. The angle of the *arrows* encodes the average number of home runs per player per game

dimension but have moved Stolen Bases from being an aesthetic to being the y dimension. League, which was on the y dimension, has been moved to be an aesthetic – *pattern*. Patterns do not have any strong natural order or metric, and so are most suitably used for a variable that itself has no natural order. League is an example of such a variable, and so using pattern for it is sensible.

It is much easier to judge the relative proportions of stolen bases using the y dimension rather than color, and, conversely, it is harder to see the spans of existence of the various leagues in Fig. 2.19. This is an important general rule; the positional dimensions are the easiest to interpret and are visually dominant. In most cases, the most important variables should be used for position; aesthetics should be used for variables of secondary importance.

Other aesthetics are better suited for continuous data. In Fig. 2.20 we show the rotation aesthetic in action. Arrows pointing to the left show low home run averages, whereas arrows to the right show high home run averages. Compare this figure with Fig. 2.17 – they are identical except for the aesthetic. Although color is more appealing, it is easier to judge relative angle and to work out which decades are above and which are below the “average” average home run rate. Color is more useful but does not allow as fine discrimination. Some authors recommend against the use of color for continuous data for this reason – or at least recommend the use of a perceptually even color scale instead of the *rainbow scale* used in Fig. 2.17, but as long as the viewer is not expected to judge relative values, any color scale that is familiar or legended well can be used to roughly indicate continuous values. I would argue (in the language of Chap. 4) that color is best used for ordinal data and effectively discretizes a continuous variable into bands of colors. If this is consistent with the goals of the visualization, then it can be used as such. If not, then better color scales should be strongly considered.

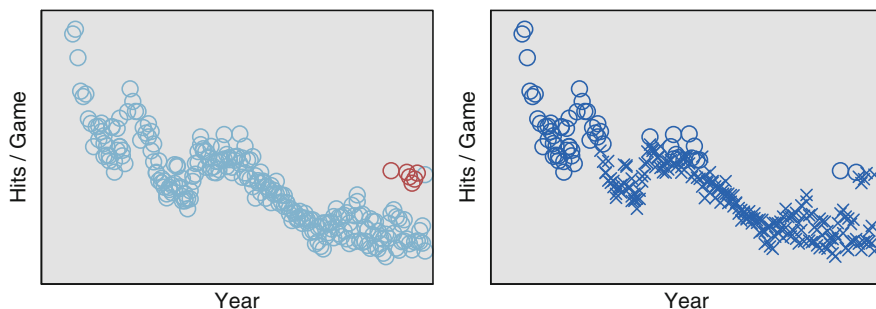


**Fig. 2.21** Baseball players' performance by decade by league. The width of the *squares* encodes the square root of the average number of runs scored per player per game, and the height of the *squares* encodes the square root of the average number of home runs scored per player per game

## 2.4.2 Combining Aesthetics

In Sect. 2.2 we discussed combining elements in the same visualization. In this section we consider combining two aesthetics in the same element. We have already seen examples of this in earlier figures where we used the size of the points to encode the number of players and the color to encode a measure of batting performance. This is a good combination of aesthetics because setting a variable to map to the size of the symbols makes that variable a *graphical weight* – the total impact of the element will be in proportion to that variable, so when we see a lot of red, for example, it is because there really are a lot of players with that attribute. Other encodings (color, shape, pattern, etc.) for that element will take up less visual room on smaller elements than on larger ones, and so are being effectively weighted by the variable being used for size. Therefore the best use of the size aesthetic is to display a variable that would make sense as a weight variable. In other words, a variable used for size should be a measure of the size, importance, count, or weight of that row of the data.

Figure 2.21 shows another combination of aesthetics. The width shows the average `RunsPerGame` and the height shows the average `HomeRunsPerGame`. Does this chart work? It is not entirely clear. It does a reasonable job of showing that the leagues are pretty similar in any given decade, but rather than focusing on the heights and widths of the symbols, the immediate impression is of the changing *aspect ratio* of the rectangles, from short and wide to tall and narrow, indicating that as time progresses, we have fewer hits overall but more big hits. If that was the message to be conveyed, then the chart does work, but if it was to show changes in each variable individually, then the chart is a failure. This highlights an important issue: Some aesthetics consist of composites of more basic aesthetics. Moreover, the base aesthetics are not independent of each other.



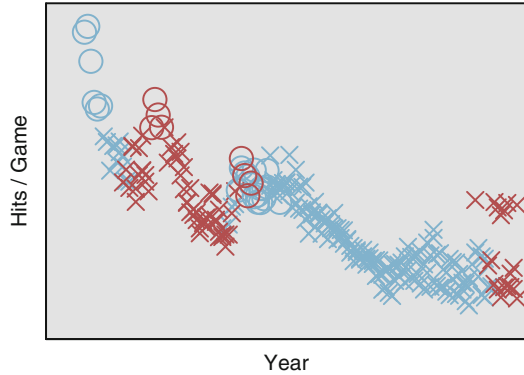
**Fig. 2.22** Separate aesthetics used on a simple scatterplot of hits/game against year. The data are statistics on average player performance aggregated by league and by year. On the *left* we highlight cases where players average more than one home run every ten games using a color aesthetic where *red* indicates a high home run rate. On the *right* we highlight cases in which players averaged more than 0.5 runs per game using shape. Seasons generating lots of runs are shown as *circles*, the others as *crosses*

Size is a composite aesthetic that can be expressed as a mixture of interrelated nonorthogonal aesthetics – width, height, depth, aspect, area, volume – and if you try to use one parameterization (in this example, width and height), then you may end up with a visualization for which another parameterization is most visually prominent – aspect in the example. A good rule when using aesthetics that combine like this is to ensure that the other parameterizations have meaning. For this figure, aspect ratio had a reasonable interpretation as the ratio of big hits to smaller hits. Area has a likewise acceptable interpretation as the geometric mean of the two variables. It may not be a commonly used measure of hitting ability, but it is at least plausible. Overall, then, this figure is truthful and informative, although possibly not for the purpose it was ostensibly designed for.

Combining aesthetics is a risky activity; it depends on the human brain's processing ability, which, unlike computers, has not been evolved for analysis and breaking down data into parts. Evolution has not suited us to the task of looking at a set of animals and understanding how size, color, shape, and other characteristics vary. Instead, we tend to see items as individuals, with all appearance details combining into a gestalt that says “*Tiger! Run!*” Fighting our brain's processing abilities is a losing proposition, and so visualizations that combine aesthetics must be carefully evaluated. We have shown how we can use color and size together effectively, but the following example shows how color and shape can interact badly.

Figure 2.22 does not need a smooth to show the general time series trend; with some interesting peaks now and again, the average number of hits per game has been decreasing over the time period 1872–2007 represented by the horizontal axis. Toward the end of the time period (1994–2007) we can see the data split into two groups – one continuing the downward trend in hitting and one group with a much

**Fig. 2.23** Baseball player performance: hits/game against year. The data are statistics on average player performance aggregated by league and by year. *Red symbols* represent seasons where players were hit by a pitch more often than one game in 40. *Circles* indicate seasons where players averaged less than 0.25 strikeouts per game



higher hitting average.<sup>8</sup> On the left we display league/year combinations with a high number of home runs in red. It is immediately obvious where those cases lie. Our brains do not need to scan through all the symbols to see that they are all located with the high-hitting group. This ability is called *preattentive visual processing*. Even if we show many millions of items, we can spot the differently colored items immediately. The right graph in the figure shows another task that can be carried out preattentively. We can spot the circles representing high average numbers of runs per game and see that they lie in three groups. Shape and color are therefore useful ways to add information to a chart. So can we do even better and use both aesthetics in a chart to make it even more useful? Consider Fig. 2.23, where we have done exactly that.

In this figure it is not easy to answer questions like: In what seasons did players have a low number of strikeouts and get hit often? That is because we have to search through the symbols and parse them in a serial fashion – there is no instant understanding. Our brains are not wired to do this form of visual recognition. Figure 2.23 is not even a hard example – we have only two colors and two shapes that need parsing. Figure 2.24 shows a much worse case, with five categories for each aesthetic. In this figure, even with only 100 data points, it is a laborious task even to find glyphs. How many symbols represent a case with `Color = E` and `Shape = 4`? How long does it take you to find a symbol representing a case with `Color = C` and `Shape = 3`?

In summary, although it is tempting to put as much information into a visualization as possible, remember that your intended viewer is a person, so you should not

<sup>8</sup>These represent players in the American League, which has a position called the designated hitter, or *DH*. Under rules in the American League the pitcher, who is usually a weak hitter, can be replaced by a player whose only job is to hit the ball; he has no defensive role at all. Rather than being surprised that this makes a big difference in hitting statistics, we might instead be surprised that this difference does not appear until 1994 – the rule had been in effect since 1973.

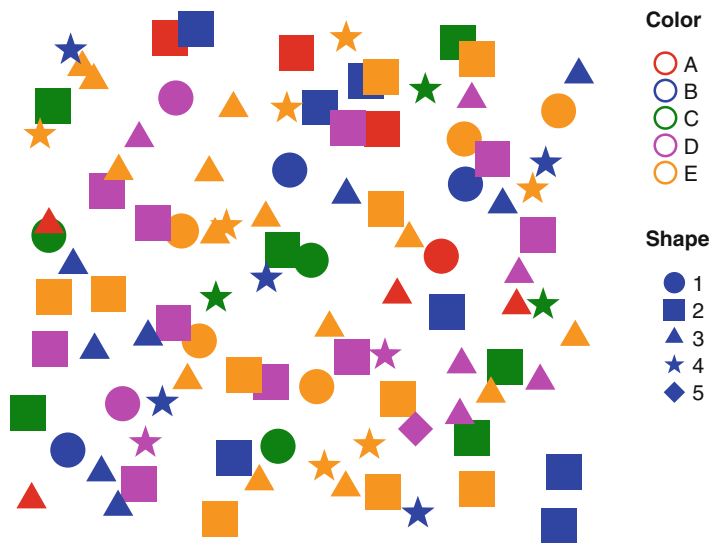
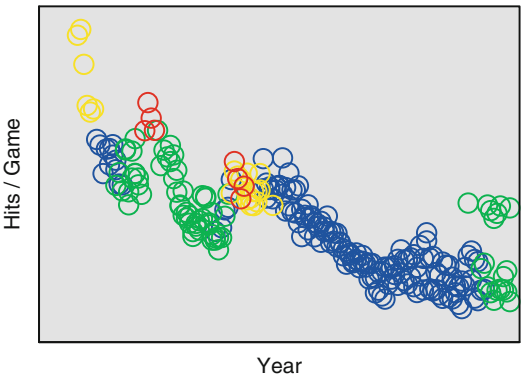


Fig. 2.24 Synthetic data set of 100 points with color and shape aesthetics randomly assigned

Fig. 2.25 A reworking of Fig. 2.23. Instead of two aesthetics for two variables, we have used color to indicate the four combinations of the variables *HBP* (HitByPitch) and *SO* (Strikeouts):

- HBP rarely, SO rarely
- HBP rarely, SO often
- HBP often, SO rarely
- HBP often, SO often



overencode your data. If you must put two variables into your chart as aesthetics, instead consider if you could use one aesthetic that encodes both variables, as in Fig. 2.25.

## 2.5 Coordinates and Faceting

Although these two topics can be discussed separately, we will deal with each of them in more detail in Chap. 5 and so will only introduce them briefly here.

Coordinates of a graph tell us how positional variables are to be used in the chart. *Faceting*, also called *paneling*, produces “tables of charts” that can be used to compare multidimensional data without needing more complex base graphs.

### 2.5.1 *Coordinates*

Examples of simple basic coordinate systems for charts include the following ones.

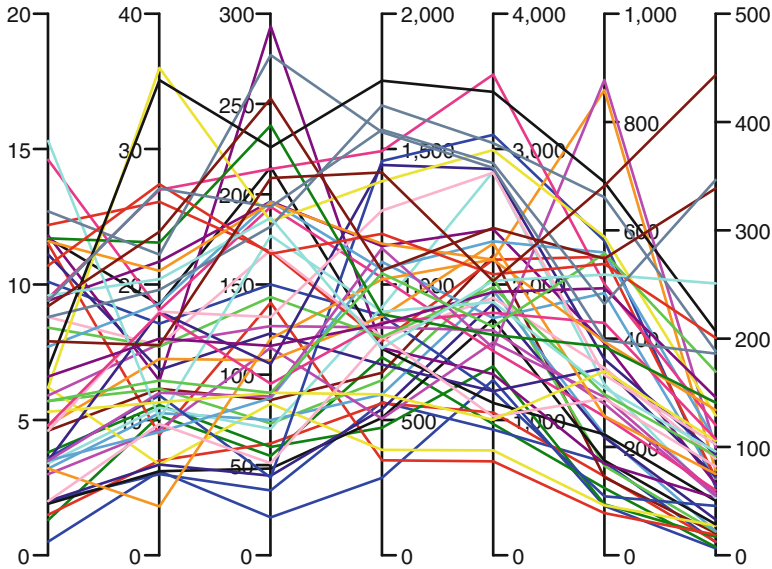
**Rectangular coordinates** form the default coordinate system, the familiar cartesian coordinates system where dimensions are orthogonal to each other. In this system axes are drawn as straight lines. The majority of charts shown in this section use simple cartesian coordinates. Typically rectangular coordinate systems are termed 1-D, 2-D, or 3-D, where the number indicates the number of dimensions they display. On computer screens and paper – 2-D mediums – we embed 1-D coordinate systems within the 2-D screen, and project 3-D coordinate systems, showing a view of them from one direction. Interactive techniques (Chap. 9) can enhance the projection to make it more natural. After all, we are used to seeing a 3-D world, so it shouldn’t be too hard to understand a 3-D graph. Beyond 3-D it gets progressively harder to navigate and gain an intuitive feel for the coordinate space, and interactive techniques become *necessary*, not simply desirable.

**Polar coordinates** consist of a mapping that takes one dimension and wraps it around in a circle. A familiar example is the pie chart, which takes a set of extents in one dimension and stacks them on top of each other, wrapping the result in a circle. In two dimensions, we use the second dimension as a radius. In three or more dimensions we can extend polar coordinates to *spherical coordinates*, in which we use the third dimension as an angle projection into 3-D, or we might simply use the third dimension as a straight axis, orthogonal to the others, giving us *cylindrical coordinates*. Mathematically, we can define these variations as transformations, where specifying a location in polar coordinates places them in cartesian coordinates at locations given by the following formulae:

$$\begin{array}{ll}
 (\phi) \rightarrow (\cos \phi, \sin \phi) & \text{Polar 1D} \\
 (\phi, r) \rightarrow (r \cos \phi, r \sin \phi) & \text{Polar} \\
 (\phi, r, \theta) \rightarrow (r \sin \theta \cos \phi, r \sin \theta \sin \phi, r \cos \theta) & \text{Spherical} \\
 (\phi, r, z) \rightarrow (r \cos \phi, r \sin \phi, z) & \text{Cylindrical}
 \end{array}$$

Polar coordinate systems have a particular value in the display of data that we expect to have a cyclical nature, as we can map time around the angle ( $\phi$ ) axis so that one cycle represents 360 deg around the circle.

**Parallel coordinates** form a counterpoint to cartesian coordinates in which axes are placed parallel to each other and spaced apart. A point in  $N$ -dimensional



**Fig. 2.26** A parallel axis plot showing crime statistics for US states. The variables being plotted on each axis represent rates of occurrence of several categories of crime. *left to right:*

*Murder, Rape, Assault, Burglary, Larceny, AutoTheft, Robbery*

space is therefore shown as a set of points, one on each axis, and these points are traditionally shown by linking them with a line, as in Fig. 2.26. In this figure, each state is shown with a line of a different color,<sup>9</sup> and this line links values of crime rates on each of the parallel axes. Parallel coordinates were popularized for visualization by Inselberg[61] with interactive techniques explored by Wegman[129] among others since.

However, to suggest that the coordinate system is a fundamental or “atomic” part of a visualization and that we can divide visualizations into useful groups based on their coordinate systems is misguided. Should we consider a 3-D view of a network as a *3-D chart* or as a *network chart*? What dimensionality is a 2-D time series chart with 3-D minicharts placed at critical points of the time series? Is a plot of event locations to be considered a *temporal chart* or a *1-D chart*? More fundamentally, is thinking of the dimensionality of a coordinate system as being of prime importance a good idea at all? Should the 1-D dotplot, the 2-D scatterplot, and the 3-D rotating scatterplot really be far apart in any taxonomy?

<sup>9</sup>Figure 2.26 shows a limitation on the use of color to denote groups. With 50 different lines, it is hard to perceive differences between all pairs of colors clearly. In this figure, the fact that each line forms its own group is the strong grouping construct – the color is added only to help the viewer separate the lines when they cross or intersect each other.

My viewpoint is that any taxonomy of visualizations is unlikely to be useful for a visualization designer. Visualization is much more akin to a language, and dividing up visualizations based on components of that language is akin to dividing up sentences based on looking at how many verbs or what types of nouns are in it. It can be a useful exercise in certain respects – for example in working out if a sentence is a question, a command, or a statement – but generally it won't help you understand what the language is really all about.

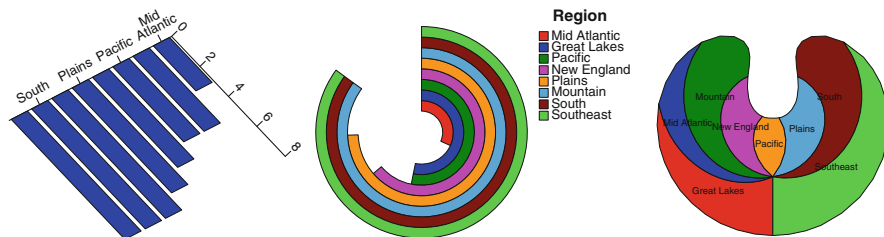
For coordinates, this is especially true. The coordinate systems described above are mathematically well defined as transformations, as was shown in detail for polar transformations. A general system for thinking of coordinates is, rather than thinking of a chart as being of a fixed type, instead to consider it as containing a *chain of coordinate transformations*. Starting with the input variables, we apply successive transformations until we finish with a 2-D output. This allows us much more freedom to apply transformations and thus produce a more powerful system. Under this formulation, we take a number of input dimensions that are taken to be in rectangular coordinates. We can then apply a chain of coordinate transformations, and the result is mapped onto the screen. Some of the more useful families of transformations are as follows.

- *Affine*, including reflection, rotation, scale, and inset;
- *Polar*, including spherical and cylindrical;
- 3-D and higher *projections*, including rectangular and oblique projections, also with parallax;
- *Map projections* such as Mercator, Robinson, Peters, Orthographic, Winkel Tripel, Lambert, and a host more;
- *Focus+context* (fisheye) transformations; these transformations emphasize designated local areas of the space and de-emphasize the rest. They are discussed in Chap. 9;
- *Parallel axes*.

One feature of a language is that it is quite possible to make nonsensical statements. It is even possible to get them published (if you are Edward Lear or Lewis Carroll). The same is true of visualizations, although it is more lamentable when the purpose is allegedly to inform, not amuse. The inappropriate use of 3-D transformations, polarizing charts when a simple bar chart is preferable – even using a Mercator projection is a poor choice when it distorts the message. The advice given in Robbins [92] is a good statement of the general advice: Since the easiest geometric comparisons for viewers to make are of lengths (preferably with all the lengths starting at the same baseline), don't make it harder on the user unless there is a good reason to. Use a simple 2-D coordinate system unless you have a good reason to do otherwise.

In the Edward Lear vein, Fig. 2.27 shows a set of three charts, each of which is best displayed as a simple bar chart. The first is an example of pointlessness – we have added in some affine transformations that do nothing to aid clarity but at least do not distort the relationship between data counts and apparent area. This is the graphical equivalent of an overly florid statement. The second chart, however, veers



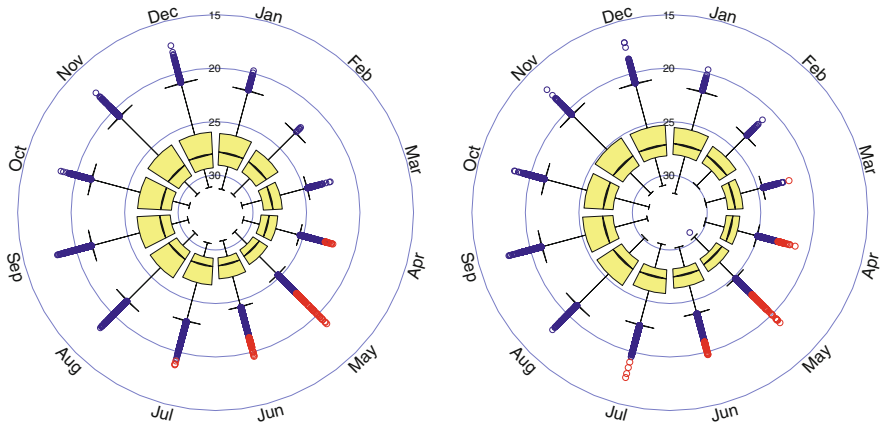


**Fig. 2.27** Three syntactically correct, but not terribly useful, coordinate chains. The first consists of a rotation transform followed by a nonproportional scaling of both dimensions. The second takes a simple bar chart and, instead of simply applying a polar transform (which would produce an at least *potentially* useful radial bar chart), transposes the coordinate system so that the bar counts become angles, massively distorting the relationship between count and area. The third one is a pie chart where we have chained a second polar transform immediately after the usual one

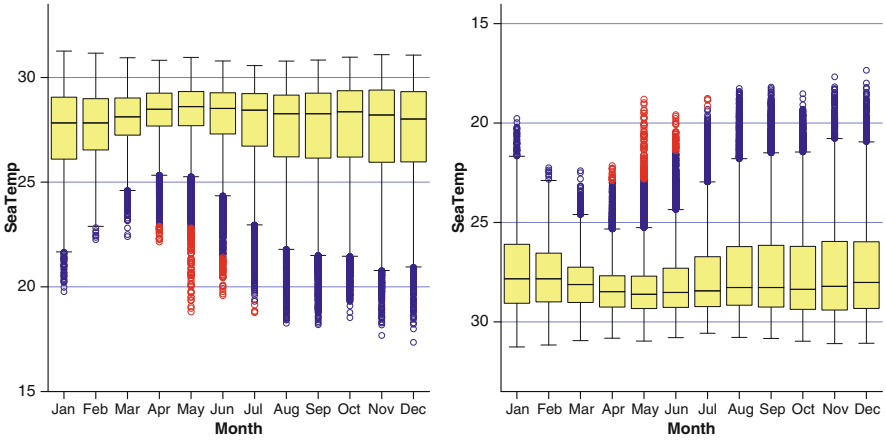
closer to evil as it distorts the data significantly. The polar transform, not appropriate for these data anyway, becomes actively bad when we transpose the chart so that the response variables (the counts) are represented by angles and the radial direction is used to show the categories. Our basically good idea of sorting the categories by count now contributes to the disaster, as the bars for the smallest counts are closest to the center and so appear smaller, whereas the bars with the largest counts are on the outside and so get a huge boost to their areas.

The final chart has an interesting history. This chart was accidentally produced by an early development version of SPSS, much to the amusement of the rest of the team. Initially we had thought of directly representing polar coordinates by a specification declaring them as  $\phi$  and  $r$  and then having the system notice that they were not simply defined as  $x$  and  $y$  and adding the polar transformation automatically. When we realized the value of chaining, we told the main development team explicitly to add a polar transformation when they wanted one but forgot to tell them to swap back to using  $x$  and  $y$  in all cases. The result was that we had all pie charts in SPSS specified as  $\phi$  and  $r$ , with a polar transformation also defined, leading to a chain of *two* polar coordinates and the third chart of Fig. 2.27. Since our general view of pie charts as a low-utility chart was well known (and is shared by Robbins [92] and Tufte [111] among many others), the initial suspicion was that we had done this on purpose; after all, was this chart really any worse than a real pie chart?

Figure 2.28 shows a better use for a polar transformation. Here the data have a natural cyclic component (the year), so mapping time around the polar angle dimension makes sense. December does follow January; there is an order. This data set consists of measurements taken from a series of buoys positioned throughout the equatorial Pacific. It is publicly available online courtesy of the UCI KDD archive [29]. Although the main interest is in studying the spatial component in conjunction with the temporal, for this figure we just consider air and ocean temperatures by month to look at seasonal patterns. As well as the obvious polar transformation, there is a less obvious one – the temperature scale, which has been mapped to radius



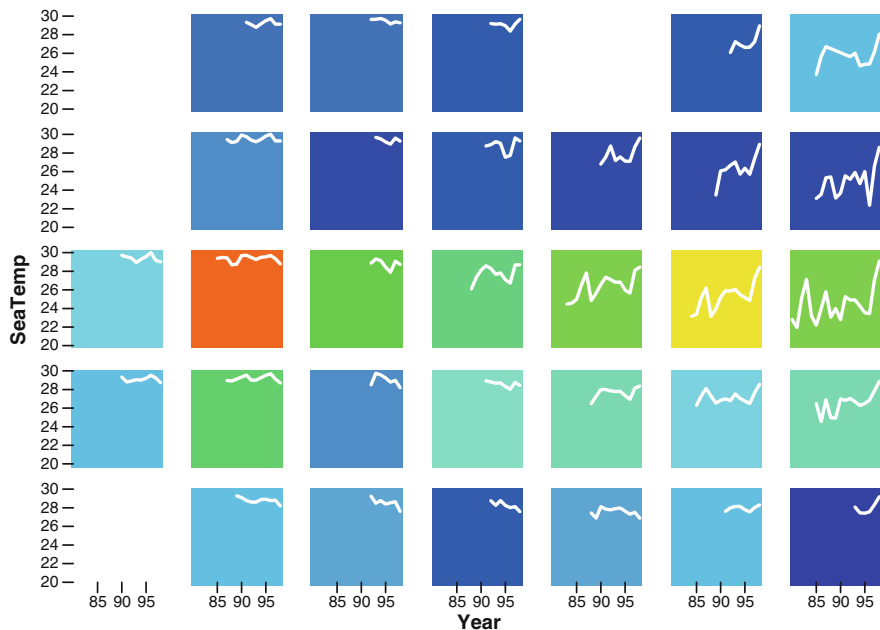
**Fig. 2.28** El Niño temperature data. These data consist of 178,080 observations of sea and air temperatures over the period March 1980–June 1998 for a number of geographic locations. In these figures we show the sea temperature (*left*) and air temperatures (*right*) for all observations, conditioning only on the month. To make the boxplots easier to see with so many outliers, we have restyled the outliers as *blue circles* and the extremes as *red circles*. Note that the temperature scale has been reversed, so that cooler temperatures are to the outside of the *circle*



**Fig. 2.29** El Niño sea and air temperatures. Two steps in the coordinate chain that finishes with Fig. 2.28. On the *left* is the original figure; on the *right* we have flipped the *y* dimension. To create Fig. 2.28, we then apply a polar transformation

and runs from *high* to *low*, with the colder temperatures on the outside of the figure. This is to highlight the outlier and extreme values for the boxplots, which would otherwise lie in the small interior area and visually be hard to separate.

Figure 2.29 shows how the coordinates for the sea temperature chart of Fig. 2.28 are constructed. We start with boxplots in a 2-D coordinate system, apply a reflection in the *y* dimension, and finally polarize to achieve Fig. 2.28. It is much easier



**Fig. 2.30** El Niño: spatial and temporal data. The faceting dimensions show the spatial locations of the buoys that collect the data. Within each facet cell is a line chart showing mean sea temperature by year. The color of the cell indicates how many observations were recorded in that cell

to compare the absolute values of medians, extents, and outliers in rectangular coordinates, but it does not give as good a picture of the seasonality. In both figures, we have changed the styles of the outlier and extreme representations to make the figure clearer. In Sect. 2.6 we talk more about the use and misuse of styles.

### 2.5.2 Faceting

Faceting is the term used in this book to describe visualizations consisting of repetitions of the same basic chart using different subsets of data for each chart. The goal of faceting is to be able to compare multiples to each other and understand what is different between them. Figure 2.28 shows two facets of the El Niño data, allowing us to compare sea and air temperatures. Faceting is also called *paneling* or *small multiples* and is a generalization of techniques like *shingling* and *trellis* displays. (See [5]; Theus [109] provides an interesting comparison between trellis displays and interactive graphics.)

Figure 2.30 shows how we can use faceting to add spatial information on the locations of the buoys. To create the faceting variables, we binned the spatial locations of the buoys into a  $7 \times 5$  grid, spanning the equatorial region in which

the buoys were observed. Within each facet, we show a time series for the average temperature per year. The background of the cell indicates which facet cells have few observations (blue) and which have many (red).

This figure demonstrates the key feature of faceting: When you facet a chart, you add dimensionality to the information you are displaying without making the base display more complex. If we added the spatial information as, say, a color aesthetic, then we would need multiple lines superimposed, one for each color.

Instead, faceting retains the simplicity and interpretability of a line chart, but with the added advantage of allowing us to compare different time series conditional on their locations.

Conditionality is another important feature of faceting. Faceting is strongest when the goal is to help the viewer compare distributions under different conditions. In this example, we can see the higher values of temperature in the “western” facets and the trend of increasing temperatures in the “eastern” facets. In general it is important to arrange the cells in the facets in a sensible order. For this spatiotemporal example, we have a clearly defined order based on spatial locations, but when we facet by variables that do not have a natural order, the results can look quite different depending on the order chosen.

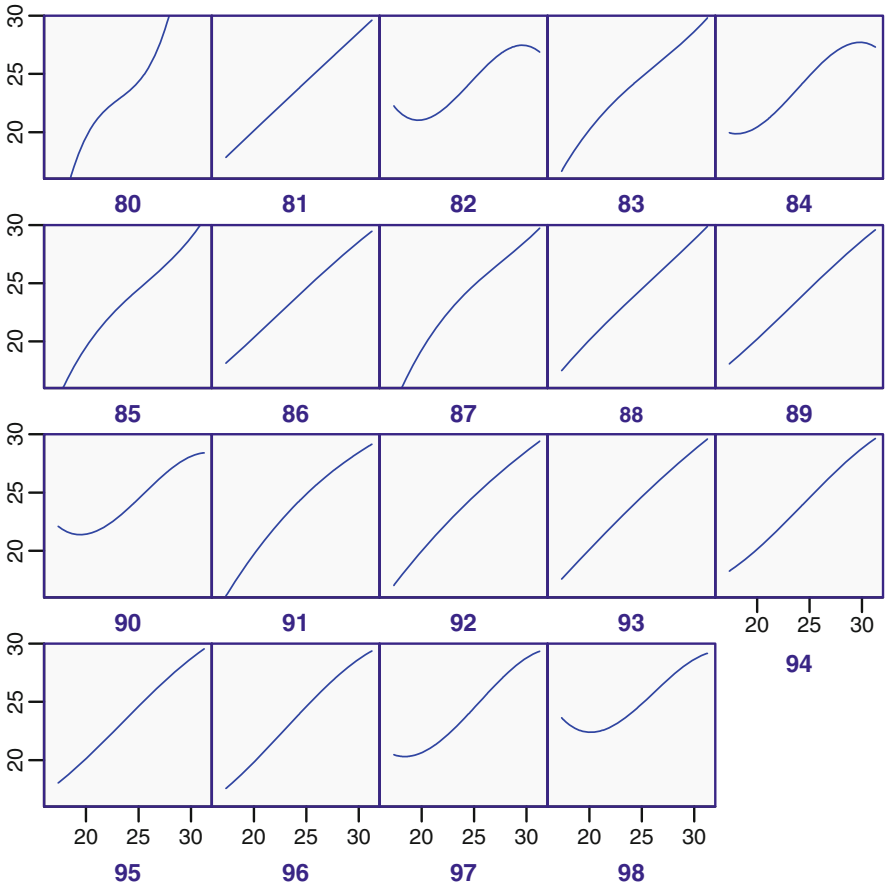
Faceting is often useful for time-based data. We can divide up the time dimension into logical chunks and use them as a 1-D facet. This is a simple but effective way of investigating changes in variable distributions and dependencies over time. Figure 2.31 demonstrates this simple usage, where we take the El Niño temperature data and facet by year. Each year is displayed as a minichart, and we can compare the minicharts to see the differences in the relationship between sea and air temperature over the 19 years in the study.

## 2.6 Additional Features: Guides, Interactivity, Styles

This section is somewhat of a “catch-all” section. In the language of visualization, this section contains the adjectives, adverbs, and other flourishes that are not necessary but add grace and beauty to a view of data.

### 2.6.1 Guides

A person who is a guide is someone who shows you the way, one who indicates interesting sights and explains details about things that might not be clear. In graphics a *guide* is part of a graph that points the viewer toward interesting aspects of the data and provides details on information that might not be obvious. A good guide is not a focus of attention; rather it illuminates another aspect of a visualization. Guides include:



**Fig. 2.31** El Niño data. A line chart showing a cubic smooth on the relationship between *AirTemp* and *SeaTemp*, faceted by year. Clear differences in the relationships can be seen. For reference, strong El Niño effects were observed in 1982–1983 and 1997–1998

**Axes:** These are guides that inform us about a dimension, showing the viewer how the data are mapped to physical locations. Advice and comments on good axes for time-based data will be given in a later chapter.

**Legends:** Axes inform about positional dimensions; legends inform about the mapping from data to aesthetics. Many of the same rules that apply to axes also apply to legends. Legends, however, can be freely moved around a chart, whereas axes are more constrained.

**Reference lines/areas/points:** These are often also termed *annotations*; they provide references and notes that inform about part of a chart’s data range. An example might be a reference line on a time axis indicating when an important event occurred; we will see examples of that throughout this book.

**Gridlines:** Gridlines consist of a set of reference lines, but positioned at locations along a dimension denoted by tick marks for an axis on that dimension. As such they share characteristics of both of the other types of guide.

In general, we will discuss guides in the context of whatever it is they are providing a guide for. The basic rule should be clear – a guide illuminates its target, so it should be as simple and self-effacing as possible to achieve that end. If the absolute data values are not of interest, then do not show tick marks. If it would be hard to tell which were reference points and which data points, do not add them. And gridlines should never be visually dominating. Follow the dictum of Mies van der Rohe: *Less is more*.

## 2.6.2 Interactivity

Interactivity is a key component of modern visualization systems. It is a rare graphical application that doesn't feature pop-up tooltips, clicking on graphical entities to see details, panning and zooming, or linked views of data. The leaders in interactive graphics are game manufacturers; the landmark game *SimCity* [142], shown in Fig. 2.32, featured all of these interactive features, and the series has gone on to improve and develop them. For comparison, one of the seminal collections on interactive graphics in the data analytic world, *Dynamic Graphics for Statistics* [23], was published in the same year. Fortunately for the dignity of statisticians, John Tukey had a head start on the subject, publishing on interactive graphics with PRIM-9 in 1975 [40] and reprinted in [27] as [39].

Interaction is a sufficiently important subject to merit its own chapter. We will study interactivity for time-based data in Chap. 9.

## 2.6.3 Styles

The use of styles for graphical displays of data is an issue that brings out strong emotions in people. Tufte [111] argues that we should attempt to maximize the ratio of *data ink* (ink used to represent the data) to *nondata ink* (the redundant ink used to elaborate or decorate the graph). While the spirit of the goal is laudable, as a measurable criterion it is not terribly useful. A time series line chart uses a very small amount of ink to draw the line as compared to the amount of ink needed to give it a good title. Is it therefore intrinsically a far worse plot than a geographic plot, since maps need a lot of ink to display their data?

Rather than focus on measures that can have little value, it is better to ask the simple question of every component of a visualization – does it help the viewer understand the data and achieve the goal of the chart? If the answer is no, then next consider if it hinders the task. If it does, then it *must* be removed. If it neither helps



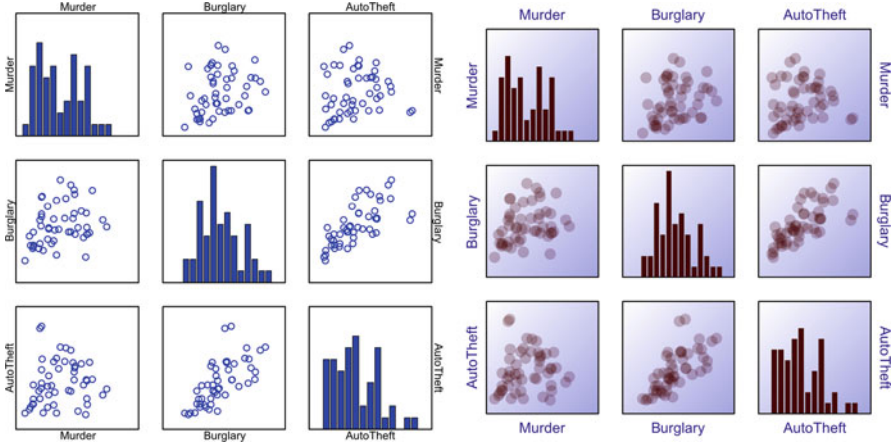
**Fig. 2.32** SimCity [142] is the name of a series of city-building simulation games, first released by Maxis in 1988 and later by Electronic Arts. It was a revolutionary concept in computer gaming – a data-heavy simulation that did not have fixed winning conditions. As part of its user interface it featured a wide range of interactive features for exploring data associated with a simulated city, including *bird's-eye overview* maps, *pop-up details on demand*, and *click to drill down*. Animated time series graphs showed key indicators of the city's progress (crime rate, pollution rate, etc.) and were *linked* to the main displays

nor hinders, then it becomes simply an issue of whether or not it makes the chart more pleasing to view.

Consider the style changes for the boxplot outliers in Fig. 2.28. They make the extremes more differentiable from the outliers, and that focuses attention on them. Since the outliers and extremes are of interest in fulfilling the goal of the chart – to understand seasonal temperature differences – the styles help the chart and are good.

Figure 2.33 shows two different *scatterplot matrices*. These are a faceting of scatterplots where each facet shows a different combination of variables. They allow you to see all relationships between pairs of variables, at the expense of requiring quite a bit of screen real estate. On the left are charts with the default styles used in this book. On the right we have overridden these default styles with some new ones, as defined by the following style sheet definition (using a definition language similar to that used by *cascading style sheets* – a ubiquitous HTML system for defining styles):

```
interval {color:#400; color2:transparent}
point {symbol:circle; size:10\%; color:#4004;
      color2:transparent}
```



**Fig. 2.33** Crime statistics for US states displayed as a *scatterplot matrix* or *SPLOM*. The plots on the *left* use the default styles employed in this book; those on the *right* use more ornate styles

```
facet majorTicks {padding:5px; font-size: 9pt;
                  color:navy}
graph cell {color:#aaf; gradient-color:white;
            gradient-angle:45}
```

The system used in this book defines styles using a syntax similar to that used in cascading style sheets [73] on Web pages. The syntax for each entry is `tag { key=value; ... }`, where the tag refers to the target item in the graphic (so “`facet majorTicks`” means the tick marks for an axis on the faceting, and the styles to apply are simply key-value pairs. Note that colors can be defined using CSS names, or as hex values, with a color of “`#4004`” denoting a color with a red value of 25% and no blue or green components, and with an opacity also of 25%.

I imagine that there will be a wide range of opinions when looking at Fig. 2.33. The simpler form is cleaner and potentially clearer. However, some may prefer the more decorated style on the right. Personally, I like aspects of both of them much in the same way I can enjoy both punk rock and opera. Contrasting the two charts in the figure, we observe the following features:

- *Font differences.* Sometimes an axis title can be relatively unimportant; if we have a series of figures with the same dimensions, or if the figure caption describes the axes, then we need not take too much space with a title. In the scatterplot matrix, the labeling for each panel is of critical importance as the order of the panels is not known a priori to the viewer. Thus, in the more ornate chart, we have made the font size larger and made them stand further out from the axis they label, increasing their visual impact.
- *Color changes.* In the left figure, the data was shown in blue and the guides in black. When displayed against a light background (the white page), this means there is a stronger contrast between the guides and the background than there is



between the data and the background. Since this is a fairly complex figure, with intermingling of guides and data, it makes sense to use a darker color to show the data and reduce this disparity. Note that we ensure that both data elements (points and bars) use the same color. This reinforces the central concept of the scatterplot matrix – that it is the same data in each panel, just different facets of the data set.

- *More prominent facet cells.* In a chart with any form of complex faceting, it is important to make the groupings into facets clear. The traditional SPLOM, which does not have any gaps between cells and which only separates the cells by a thin line, confuses the viewer by making it easy to see groups or clusters across cells, instead of within cells. Separating the cells helps, and using styles to add a light background helps more. The gentle gradient color on the cells is purely decorative. If you or your audience prefers a more austere look, leave it out.
- *Opacity.* Also known by its inverse, transparency,<sup>10</sup> this is being used as an alternative to the intersecting open circles to solve the problem of occluding circles. Overlapping shapes can clearly be seen, and as more objects are drawn on a given location, that location becomes darker, effectively giving us a “poor man’s density estimate.”

There is no overall right or wrong in adding styles to charts. If in doubt, it is probably safer to do less rather than more, and so the general rule would be to prefer simpler charts. Sometimes style additions can enhance the utility of a chart. If so, the additions should be made. If, however, they are merely for decorative ends, it becomes a question of the desires and preferences of your intended viewers. Like music or painting, there are many rules and guidelines that might be made, but in the end, the proof is in the finished product.

## 2.7 Summary

This chapter has been an introduction to the language used in this book to describe visualizations. It has also provided a framework for how visualizations are put together, using the metaphor of a language rather than a taxonomy of choices. Using examples, it has introduced some general concepts and offered advice applicable to a variety of data types and situations. Although the uses for time-based data have been noted, the main focus of this chapter has been on the overall framework and general application of visualization. In the following chapters we will delve more deeply into visualization specifically for time-based data. We will start by talking about the data.

---

<sup>10</sup>A 100% opaque object is 0% transparent and vice versa. Using opacity is preferable as the more opaque something is, the more visually prominent, and so opacity can usefully be used as an aesthetic when mapped to counts, weights, sums, and the like, since larger values of these statistics should be shown more prominently.

## 2.8 Further Exploration

Wilkinson's *The Grammar of Graphics* [135] is the fundamental book on this framework, but several other frameworks have built on Wilkinson's ideas, including *ggplot* (described most fully in [131] with an earlier reference in [130]), which is an R implementation of many of the ideas in the grammar. Tableau Software [104] is a successful commercial visualization company that has focused on the faceting aspects of the grammar.

For each individual topic in this chapter there are many examples and avenues of exploration, which are detailed in the subsequent chapters. There are also many descriptive frameworks that classify visualizations using various splits (by dimensionality, base data type, or interactivity, for example) but little else that describes a generative framework that can be used to construct as well as merely describe charts. Some charting systems have provided some composability features. The Web graphics framework Dojo, for example, has an extension, *dojox.charting*, that provides composability and uses many of the same basic chart components. It is best explored via the Web, but the latest version of standard references ([93], for example) should be updated to contain sections on the charting capability.

Visualizing Time

Designing Graphical Representations for Statistical  
Data

Wills, G.

2012, XVI, 256 p., Hardcover

ISBN: 978-0-387-77906-5