Statistics and Computing

Graham Wills

Visualizing Time

Designing Graphical Representations for Statistical Data

Springer

Statistics and Computing

Series Editors:
J. Chambers
D. Hand
W. Härdle

For further volumes:
http://www.springer.com/series/3022

Graham Wills

Visualizing Time

Designing Graphical Representations for Statistical Data

Springer

Graham Wills
Hidden Spring Dr. 1128
60540-4112 Naperville, Illinois
USA
graham@spss.com

ISSN 1431-8784
ISBN 978-0-387-77906-5 e-ISBN 978-0-387-77907-2
DOI 10.1007/978-0-387-77907-2
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011940977

Although this book contains tributes to famous men and women who have invented unique and novel visualizations, and to experts who have distilled knowledge and advanced the science of information visualization, this book is dedicated to those people who designed reports, published charts, and created visualizations and were not content to use the defaults but instead took the extra effort to make their work more truthful, more beautiful, and more useful.

Thank you!

Preface

Art or science? Which of these is the right way to think of the field of visualization? This is not an easy question to answer, even for those who have many years of experience in making graphical depictions of data with a view to helping people understand them and take action. When we look at beautiful hand-drawn pictures of data, carefully composed by talented individuals, we are drawn to the artistic side. In some ways those charts are discouraging; their artistic elegance implies that the creation of good visualizations is not an option for most of us. There are books that provide rules and advice on how to draw graphs. Some give general advice, suggesting that such and such is good, but this other is bad. Others give specific advice such as requiring all charts to have a title or all axes to go to zero, but these are often tied to specific visualizations and so are not general enough to qualify as scientific principles. They are valuable for describing existing visualizations, but not general enough to provide guidance for future visualizations. If you are designing something new, advice on a bar chart is not especially helpful.

In this book I want to bridge the gap and not simply give rules and advice but base these on general principles and provide a clear path between them, so that the rules and guidance fall into place naturally, due to knowledge of those principles. In terms of the art/science split, I want to advance the scientific component. There are excellent books describing artistically superb plots; however, my goal is not simply to be descriptive, but to be *prescriptive* – to allow people to start with a goal in mind and design a visualization that fulfills that goal clearly, truthfully, and actionably. Because I have an essentially scientific direction in mind, I will concentrate on reproducibility. A chart that is wonderful for exactly one data set is of little interest. It can be appreciated and enjoyed, but the important question must always be: What can I learn from this graphic that I can apply to other data? With this in mind, the examples in this book have been chosen to be realistic rather than exemplary. I have made a definite attempt not to choose data that make a picture look good, but rather to choose data for which a chart should be applicable. If the result is not perfect, I prefer to present imperfection and explore remedies rather than look for a different data source.

This book is concerned with the graphical representation of *time* data. Time is *special* – it doesn't behave quite like other variables. It has an inherent direction and determines causality. Time can be recorded in many ways: it can be linear or cyclic, categorical or continuous. Even the written format of a piece of time data can be curiously baroque; mixtures of words, numbers, and special symbols make up the time "Monday the 13th of October, 11:45 am." What other form of data might occur in so obscure a format? All data are recorded at a certain time, and so all data have a time component, even if it has been removed or deemed a priori as uninteresting. This makes time data both unique and universal, so understanding how best to portray them not only is challenging but has wide applicability.

The portrayal of time data is ubiquitous. Any newspaper will feature time-based plots; any company report will show historical data as charts. Even the gas bill for my home invites me to compare a time series of the cost of heating my home against one of average monthly temperature. Because of this generality, I have written this book to cover a range of different users. A visualization expert designing tools for displaying time will find it valuable, but so also should a financier assembling a report in a spreadsheet or a medical researcher trying to display gene sequences using a commercial statistical package. You have data, you have a goal in mind. Now all you need are the tools to graph the data and so achieve the goal. Read on!

Graham Wills

Acknowledgements

The only way to know the effort needed to write a book is to do so yourself, and only authors know the debt of gratitude they owe to others. Warm thanks are due to many people, broadly classified as shown in the diagram below. Any errors and mistakes within the book are entirely my own.

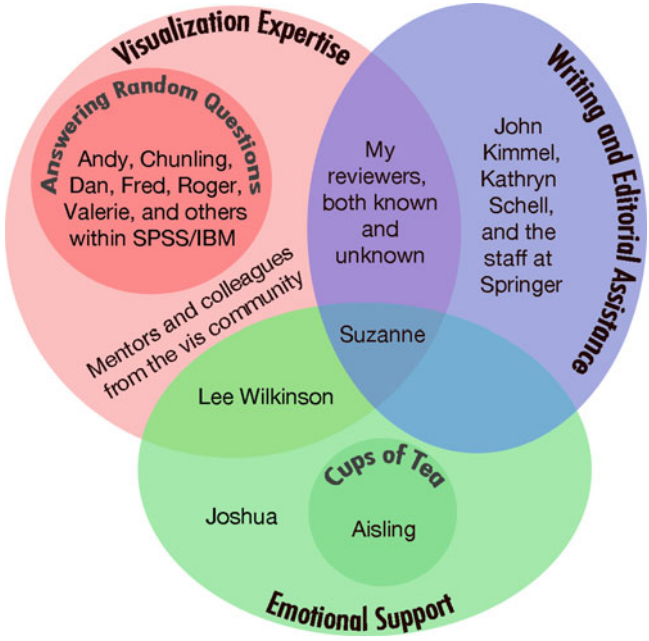


Fig. 1 A modified Venn diagram showing where acknowledgement is due; it shows the major sources but does not include everyone who has helped or influenced my thinking or who has taught me or argued with me over the years. The sum total of those contributions would be large; as this book will indicate, it is often small contributions that make or break any major endeavor

A Note on the Figures

One of the advantages of working for a company, rather than in an academic position, is that you get a different perspective on how visualizations are used. That turns out to be "every way you might ever think of, and then some." Once your algorithm for generating histogram bin widths has been used by tens of millions of people, on hundreds of thousands of data sets, in over a hundred countries, and any time it didn't work out for them you got a note to that effect, you start to appreciate the value of robustness not just as a statistical quality but as a *visualization* quality. Now, whenever I see a cool new technique being presented or see an algorithm described that works on one form of data, I immediately think: Will that work in general, or is it too fragile?

Not only is this a pervasive theme in the book, but it has also motivated the design of the figures and, in particular, the lack of postproduction editing. It has been very tempting to take some figures in the book and import them into a graphics editing environment and do a little subtle rearrangement or improvement. Nothing dramatic – just the sort of thing that magazines do to their cover pictures; smooth over imperfections, thin down the bulges, make something already attractive even more so. Several of my reviewers indeed recommended I do exactly that.

Instead, I have chosen to leave the outputs unedited. The charts I have created (all those in the main chapters that have not been attributed to others) are straight from production. I have used the *VizML* language (an XML specification language) to create chart specifications, and the output in the book is exactly what came out of the package. VizML is a basic language used in virtually all output of SPSS (now part of IBM) and is available and usable by anyone who owns the major SPSS products. In fact, all the VizML figures in this book were generated automatically by a set of of Python libraries I wrote that encapsulated common actions for editing VizML specifications. As an example, Fig. 3.6 on page 72 was generated by the following Python fragment:

```
Movies = Datafile('Movies.csv')
thin = 'element{stroke-width:0.25px} visualization{margin:5mm}'
T.Histogram.make(x=Movies.Sales, name="MovieSalesA") \
    .remove(Axis,1).addStyle(thin).outputSize("4in", "3in")
```

```
T.LogDistributions.make(X=Movies.Sales, name="MovieSalesB") \
    .remove(Axis,1).addStyle(thin).outputSize("4in", "3in")
```

I used a standard histogram template (`T.histogram`) for the first part of the figure and a template I designed myself for the second one. For each chart I killed the *y* axis, added some styles, and set the output size. To generate the figures for the book, I kick off a Python script, wait 10 minutes, and have all my figures.

That is my goal: not to present a set of graphics that are as good as any designer could produce, but instead to produce graphics that a data professional could create to solve a task. Visualization is a tool, and I want a tool that *works*.

Contents

Chapter 1
History

HISTORY, n. An account, mostly false, of events, mostly unimportant, which are brought about by rulers, mostly knaves, and soldiers, mostly fools.

–Ambrose Bierce, The Devil's Dictionary (1881–1886)

1.1 The Importance of Time

Measurement and recording are part of the scientific approach, no less for time than for any other form of data. (Chapter 4 discusses details of how we define data and ways to describe time in detail.) The history of how those measurements have been made – even the words that have been used for units of time – makes for a fascinating study. For example, Brian Hayes describes the astronomical clock of the Strasbourg Cathedral in his essay "Clock of Ages" (reprinted in [53]). This is a mechanical clock built over 160 years ago that measures in addition to the usual values:

- Sidereal time (measured by the Earth's rotation),
- Local solar time (where noon is when the sun is highest),
- Local lunar time,
- A counter for the years,
- The current date, including leap year calculations, and
- The dates of movable church feasts, including Easter (a complex calculation, only standardized in 1582 by Luigi Lilio, as described in [81]).

This entire system works with gears, from fast-turning ones to a small gear that turns only once every 2500 years. The Strasbourg clock is a tribute to the importance of being able to calculate and measure time in multiple ways.

Although vital to the smooth running of civilization, calendars and the measurement of time have not always been standardized. Holford-Strevens [56] introduces and describes the major different calendar systems that have been employed across

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_1,

the ages; the major division into lunar and solar calendars and how various systems attempted to resolve the differences between them. He gives the histories of the Babylonian, Egyptian, Jewish, Roman, Julian, and Gregorian systems and their evolutions. Today, virtually all countries use either the Gregorian or Revised Julian calendars, which are not due to differ until 2800,[1] so dates under modern notation can safely be assumed to be comparable.

The *accurate* measurement of time was another historical goal with important applications. In the eighteenth century, accurate measurement of time was necessary to be able to calculate a ship's longitude with precision. This was considered important enough that million-dollar prizes (in modern-day dollars) were awarded and ship owners were prepared to pay amounts up to a quarter of their ship's value in order to buy a highly accurate clock. How well did they manage? In 1761 John Harrison's H4 clock made the trip by sea from England to Jamaica and lost only five seconds on the voyage [103]. Pretty impressive.

Although the history of measurement and recording of time is a fascinating study, in this book we are concerned primarily with the display of time. In line with our aim of making *informative* visualizations, this chapter will take a look at a set of historical visualizations with the goal of learning what made them useful and what lessons we can apply to our visualizations.

1.2 Ancient Visualizations of Time

Figures 1.1 and 1.2 show a pair of petroglyphs (also known as "rock carvings" or "rock engravings" that were created probably a couple of thousand years ago near the area of the Grand Canyon in Arizona, USA. They are not particularly notable or exceptionable specimens; in fact, they portray a fairly common pattern of markings that experts believe may represent early calendars (or, at the least, records of past time periods). The earliest forms of data recording consist of notches or scratches representing counts of various qualities, and so the representation of time as a sequence of counts of units past a known event is a natural first step in recording time.

Early historical documents often refer to dates in a similar manner, for example:

- "In the four hundred and eightieth year after the Israelites had come out of Egypt, in the fourth year of Solomon's reign over Israel, in the month of Ziv, the second

[1] 2800 is a leap year in the Gregorian calendar, but not in the Revised Julian calendar. Under the latter system, leap years for centuries only occur when the year divided by 900 has remainder of 200 or 600. This system is more accurate than the Gregorian calendar in the (very) long run, requiring less adjustment to match the actual year length.

Fig. 1.1 Petroglyphs near the Grand Canyon. These cave carvings date from as far back as 500 B.C. and were found at a site believed to be used as a temporary hunting camp. Although this particular image looks somewhat like the skeleton of a fish, there are many similar diagrams that look more tabular. These images are often thought to represent a "hunting calendar"

Fig. 1.2 This figure shows another "calendarlike" petroglyph found near the Grand Canyon. Beside it is an iconic representation of what appears to be a volcano. If so, a natural interpretation is that these boxlike divisions indicate periods of time since a significant event

Fig. 1.3 Stonehenge may be the most famous ancient monument in Britain. For over 850 years, researchers have searched for explanations of its purpose and why and by whom it was built. Photograph ©Mark Radford 2009, used with permission [89]

month, he began to build the temple" – an account of the building of Solomon's Temple, taken from the Book of Kings.[2]

- The traditional Japanese calendar consists of eras based on the reign of the emperor. This calendar is still used in several official contexts – the Japanese patent office only changed from this dating system in 2002, or "Heisei 14" as that year is called in the traditional system.
- The terms "D-Day" and "H-Hour" are used in military operations to designate the time an operation begins. Times are measured relative to this event so that "H-3" means 3 hours before the operation commences and "D+7" a week after its commencement. This convention has been used since at least 1918.

From a small, common example of prehistoric creation we leap to one of the most famous sites in the world: a collection of large stones, carefully arranged in a set of concentric rings – Stonehenge. Evidence makes clear that it was built in successive stages over a period of several hundred years, and considerable effort was put into its creation, so there must have been a strong reason (or reasons) to have it built.

[2]The phrase "in the *N*th year" of someone's reign gives historians some trouble as even within a single source it may be used in different ways. If a king starts ruling in a given year, sometimes that is described as "the first year," but other recorders may use that term only for the first *full* year, using a phrase like "the year when so and so became king" for the partial year of ascension.

The stones have been placed with a precision of up to a few centimeters, leading to several theories for this concern for accuracy. One theory, dominant for the last 200 years, is that the site is a giant astronomical observatory or calendar, designed to allow its users to observe or measure seasons, lunar cycles, and similar time-based occurrences.

Despite the lack of strong evidence to back this theory up,[3] the theory clearly has been compelling. It is both a cautionary tale in trying to guess the motives of other people and a statement about the importance that we ourselves give to time. In essence, we think that if a prehistoric people were to go to such great lengths to make such a structure, it must be for a very important reason – and what is more important than time?

1.2.1 Summary

The major lesson we can learn from these ancient representations is that *t*ime is important. Many representations of time exist in the ancient world, and even when we are unsure about an artifact's importance, we can see that some features are time related. Since our goal is to make useful visualizations, it is important to remember that these visualizations were created for a purpose. We should do the same, creating visualizations that serve a purpose. Chapter 3 describes a design methodology that accomplishes this.

These depictions of time show how time is often *d*ivided into discrete units; calendars divide time into days, months, and years – although we speak of time flowing, in practice we often divide it up into chunks. In Chap. 8 we address this technique and use "chunked time" throughout the book in an informal way. Relatedly, time can be recorded as an ordinal variable – a set of ordered categories – that corresponds to the way calendars are laid out. Chapter 4 deals with the various ways in which we can consider time data. One such way is to consider *durations*, which can be defined as the time elapsed since fixed events.

In Chap. 6 we discuss *a*xes – guides that show time dimensions and allow us to identify what time a region of the chart represents. The main axis of Stonehenge is a literal axis being used to represent the time dimension. For these examples, measurement might be the most important feature, but for all time visualizations we must be able to answer the question: When did this interesting feature happen? Guides have a particular significance for time data, as measurements are often made relative to known events. In designing visualizations, adding known reference times is highly advisable.

[3] Johnson [64] gives strong arguments that the only real calendric detail is the alignment of the major axis along the line of the summer–winter solstices and that an alternative explanation should be sought for other features.

Finally, the use of a *p*olar coordinate system is seen in many ancient depictions of time. The sun travels around the world, the world rotates around the sun, and so many of our systems for measuring time are cyclical in nature; sundials must use a polar system for laying out the hours, mechanical clocks have dials, and so on. Not only is this needed for mechanical reasons, but using a circular coordinate system allows us to visualize data that may have a cyclical nature. Many phenomena have obvious dependencies on natural cycles (weather patterns, for example), and so a final lesson to learn is that we should not restrict ourselves to portraying time only as a linear dimension.

1.3 Playfair

William Playfair has been credited with the invention of the time series chart and in 1786 published a book, *Commercial and Political Atlas* ([87], republished as [123]), that introduced a number of new visualizations to the public, including many charts showing time data. Playfair was a strong character and is the focus of the first part of Wainer's book [122] on graphic revelations. As an example of what Playfair was like, I offer up the title of Chap. 3 of this book: "William Playfair: A Daring Worthless Fellow." His worth in the realm of statistical graphics, however, is unquestioned.

Playfair made a number of strong statements in his introduction in favor of graphical representations, of which the following is representative:

> Figures and letters may express with accuracy, but they can never represent either number or space. A map of the river Thames, or of a large town, expressed in figures, would give but a very imperfect notion of either, though they might be perfectly exact in every dimension.
>
> In an affair of such consequence, as the actual trade of a country, it is of much importance to render our conceptions as clear, distinct, and easily acquired, as possible . . . A man who has carefully investigated a printed table finds, when done, that he has only a very faint and partial idea of what he has read.

Figure 1.4 is a reproduction of one of Playfair's figures. The Annenberg Rare Book and Manuscript Library at the University of Pennsylvania provided the original and assisted in recovering a high-quality reproduction of the original for [123], which has been reproduced here. The figure is intended, as stated by Playfair, to show the balance of trade clearly and succinctly. Playfair's charts mark a major milestone in the history of data visualization, and for time visualization in particular they can be regarded as the first quantitive charts of time.

Playfair's charts show several *e*lements in the same graphic. Here we have lines for both imports and exports and the area element that is defined by their difference. Further, the lines are given unique colors to distinguish them, and the areas are colored depending on the sign of the difference in values.These are examples of the use of *a*esthetics, which take a basic chart and add information without changing

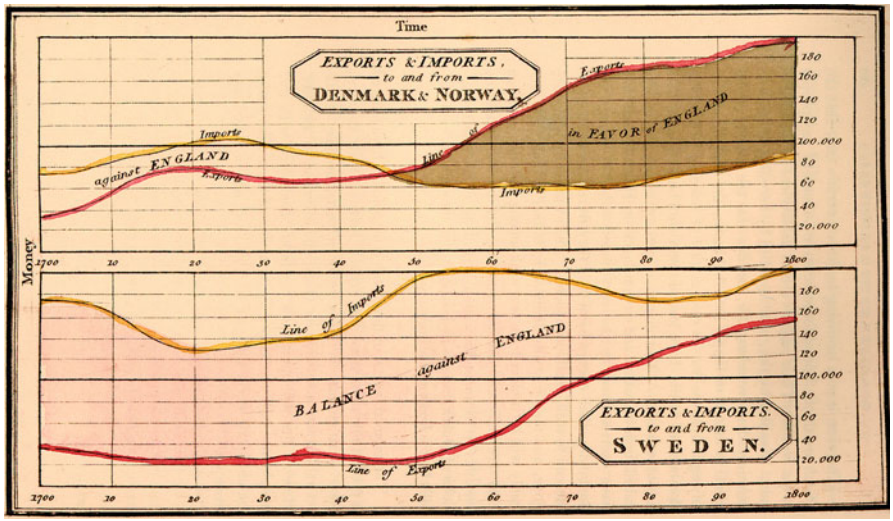


Fig. 1.4 This figure shows two graphs: the balance of trade between England and Denmark/Norway and the balance of trade between England and Sweden. In both graphs there are two lines, one for imports and one for exports (from England's point of view). The area between the lines has been colored to indicate the direction of the balance. The original of this figure is in [87], reprinted in [123]

the structure[4] and are described in detail in Chap. 7. Playfair uses multiple elements in other charts, including a bar and line chart with different vertical dimensions. He also shows stacked areas and uses patterns and colors for aesthetics on bar charts as well as areas. And it would be remiss not to mention that he also invented the bar chart – the first depiction of data in a purely abstract coordinate system without either dimension representing space or time.

A second innovation is the use of an *area* to represent the trade amounts. As we will see in Chap. 2, an area is a good choice when the values being represented on the vertical dimension are additive. We can add money, so an area makes sense. Further, the size of the slice of area between the two vertical lines represents the sum of all exports in the time period between those lines, so we can make comparisons of total area.

The charts are also *aligned*; the time values are aligned vertically between the two figures and their vertical axes span the same range, making it possible to do area comparisons accurately.This technique, here applied to two charts, can be applied to many charts using a technique termed here *faceting*, elsewhere called "paneling," "small multiples," and "trellis." In Fig. 1.4 we can directly compare the

[4]In this book the word *aesthetic* is defined as a mapping from data to graphic attributes of a chart element, such as color, size, pattern, opacity, etc.

areas between the two charts and also compare the areas "in favor" and "against" in the upper chart and make correct judgments based on them.[5] Faceting is studied in detail in Chap. 6.

As if this were not enough, Playfair also gets the details for his *a*xes right. Gridlines are clear, but not prominent, with major gridlines more heavily weighted than minor ones. His labeling of the tick marks drop unneeded precision and are placed at appropriate tick locations. In Chap. 6 some of these details are discussed, and a brief look at the default output of most charting packages will indicate that the creation of a good axis is not an easy task.

We do not expect to see any interactivity in hand-drawn charts from the 1700s, but Playfair does lay the groundwork for several key interactive techniques. Consider Fig. 1.5 (plate #1 in [87]). This is an *overview chart*, summarizing the trade between England and all its trading partners. One interesting point to note is that the chart shows a focus on the years from 1760 onward. The lines for both imports and exports are less smooth, leading us to believe that there are more data available in this period, and Playfair has given us minor gridlines only in this time period – a static technique that focuses attention and helps us concentrate on this part of the chart. In Chap. 9 this technique is expanded to a fully interactive context; the use of *distortion* techniques that selectively magnify part of the axis is demonstrated. Playfair paves the way for this very recent interactive technique with the selective use of guides to focus our attention on this area.

1.3.1 Summary

Playfair establishes the basic structure of time series charts with these graphics. Most important is the basic structure – a *two-dimensional* coordinate system with time running horizontally and a quantitative *y* dimension. This basic structure, *time* $\times$ *quantity*, forms the standard framework for a time series chart and is used extensively throughout this book. His use of multiple elements, aligned axes, and axis drawing techniques is also an important design development, which we will explore in the course of the book.

A final point to make is on the presentation of the charts themselves. Playfair structures the book by giving us an overview chart as the first figure and then moving on to show details of individual countries' trade in subsequent chapters. This is another important interaction technique – start with an overview of the data, then filter by a relevant variable (in this case, the country being traded with), and show details on those drill-down charts only on request. In this static medium, the details

[5]Up to a point. Wainer and Spense indicate in [123] that Playfair may not have been as accurate with his depictions of the data as he could have been. The method is correct, but the execution seems to be a little haphazard.

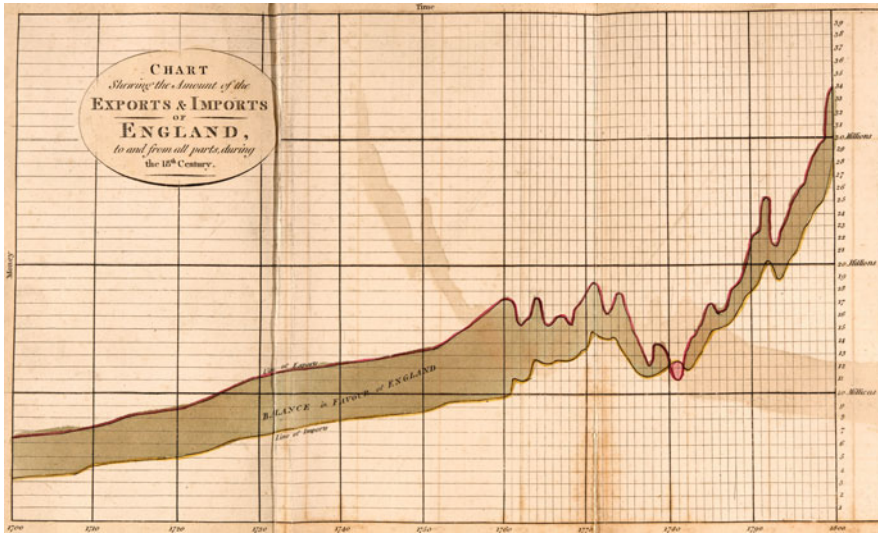


Fig. 1.5 This area graph shows the overall balance of trade between England and all countries with which England traded over the period 1700 through 1800. Most notable in the graph is the single dip "into the red" that occurs between 1780 and 1782. At this time England was heavily involved in fighting the Revolutionary War in North America, as well as a couple of conflicts in India; the first Anglo-Maratha War, fought between the British East India Company and the Maratha Empire in India; and the second Anglo-Mysore War, started opportunistically by Great Britain on the grounds that since they were already at war with France (who joined the Revolutionary War on the US side), they might as well have a go at kicking them out of India, which was a far more important possession in terms of economic value. Spain joined the war (as an ally of France) and the Dutch also joined in, with Britain declaring war on them in 1780 to start the Fourth Anglo–Dutch War, which worked out very well for Britain, as they took a number of Dutch colonies. This graph shows the effect of all these wars on the balance of trade; the next important step would be to *d*rill down and see how trade with each of the opposing countries was affected. The original of this figure is in [87], reprinted in [123]

are the body of text in the chapter. A famous basis for visualization systems, the "Visual Information-Seeking Mantra" of Shneiderman [98], states:

> *Overview first, then zoom and filter, and finally, details on demand.*

Playfair, 210 years earlier, used this basic organization principle for his exploration of trade balances, and this approach is described further in Chap. 3.

1.4 Napoleon's March

In this section we will look at a chart that is widely regarded as a classic of information design, Minard's map of *Napoleon's march on Russia*, reproduced as Fig. 1.6. Charles Joseph Minard was an engineer who created many innovative

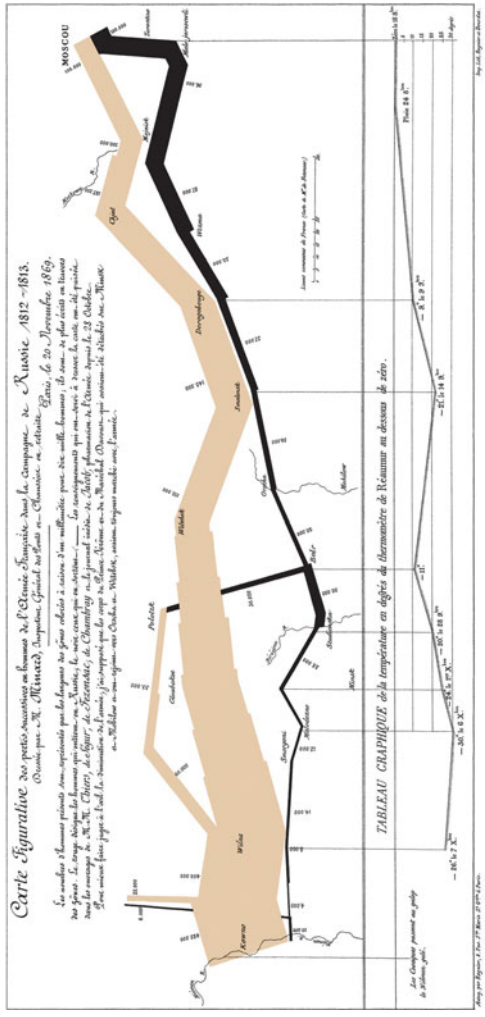


Fig. 1.6 This figure is titled "*Figurative map of the successive losses in men of the French Army in the Russian campaign 1812–1813*" and was created by Charles Joseph Minard [79] in 1861 to show how large a disaster Napoleon's march into Russia was in terms of human life. The upper part of the figure shows the path of the army's advance and retreat, with the width of the path being proportional to the size of the army. It is superimposed over details of the terrain crossed. The lower linked chart shows the temperatures on the return journey

and, more importantly, functional graphics that bear careful study. It is well worth searching out other examples of his work and analyzing them; in this section we will restrict ourselves to his most famous work.

This chart is not a simple one; in fact some have argued that since the main message is an essentially obvious one ("98% of Napoleon's army died!"), a basic bar or pie chart with two values, one for deaths and one for survivors, would do as well at conveying the same message. However, the other pieces of information that are present provide context, answer related questions, and prompt exploration. The chart is built on several sets of data:

- *Geography*: The main chart is a map and depicts a selection of rivers, cities, and battles. The path showing the advance and retreat is located using map coordinates.
- *Path*: The path of the army is drawn directly on the map and is color coded by direction: gold heading into Russia, black retreating out.
- *Count*: The number of soldiers is represented by the width of the path, from 480,000 at the beginning to 10,000 at the end.
- *Temperature*: For the retreat only, the air temperature is given at selected points along the journey, represented by a line chart at the bottom, with thin lines linking the two charts.
- *Time*: Time runs right to left as the army retreats, and the line chart at the bottom gives dates at several locations.

The geographic information is essentially a guide in this display – a background that gives context and helps us understand the main effects – for example, showing that when rivers are crossed in cold weather, people die. The rest of the data can be thought of as a single table with each row being a location along the march, and the variables being *latitude*, *longitude*, *count*, *temperature*, and *time*.

This figure has a lot to live up to. Tufte [111] declared that it "may well be the best statistical graphic ever drawn," and it has received similar accolades by other authors.In this book our goal is not so much to analyze charts for their unique beauty and individual utility, but to understand how useful the chart is and what general lessons we can learn from it. The original was hand-created, but numerous versions of it have been created, some in an attempt to remain true to the original, some more radical reworkings. Several are detailed in [45] and collected on the Web at [44]. The version in Fig. 1.7 was created using the *VizML* visualization system used throughout this book. Only the cities have been retained from the geography, and more prominence has been given to the temperature chart, since our goal is to emphasize the time component more than Minard originally did.

1.4.1 A Fortunate Correlation

In statistical analysis, if we are trying to predict a target variable, and two of the variables we are using as predictors are highly correlated, then this can often cause problems. In statistical graphics, however, it may turn out to be a good thing. I would argue that Minard's chart is a great chart because it is based on a highly successful and simple time-based chart that has a fortunate correlation with geography.

Refer back to Playfair's figures on balances of trade (Figs. 1.4 on page 7 and 1.5 on page 9). They are time series charts that display values of a variable using an area, where that variable is a measure of the size of something – a display very similar to the paths in Fig. 1.6. Minard's chart works because it shows a time series of army size, with the *time* variable replaced by the highly correlated *longitude* variable. That

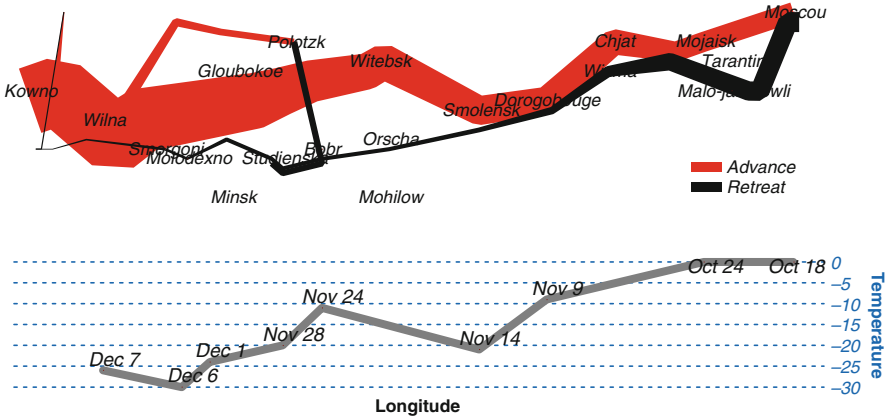


Fig. 1.7 Napoleon's March on Russia. This figure is a reworking of Fig. 1.6, simplified to allow it to fit into a smaller space and remain legible. It also has allocated equal space to both the map and the temperature graphs, making them equal parts of the story

allows the time series of army size to be naturally integrated into the map display without needing any explicit linking. This has a few consequences:

- One potentially interesting factor – how fast the army advanced – is not available to us. The temperature graph that is linked to the main chart helps us infer the pace of the retreat by linking locations to a line chart with time annotations. We would need another linked view for the advance.
- The area of the path and the rate of diminution show the army size per mile of ground covered, which is a somewhat unusual and less informative quantity than would have been shown using undistorted time for the horizontal axis. In that case we would have seen the rate of change per day. However, it could be argued that since the goal was to get to Moscow at all, rather than in a given time period, base units of distance traveled make just as much sense.
- Since for the complete march the longitude does not correlate well with time, Minard had to split the chart in two, one path for the advance and one for the retreat. Each is dealt with separately.
- For the retreat, which is the most recognizable "time series" chart on the bottom, time runs right to left, which is disorienting as even in cultures where text might read in other directions, time is almost always shown running left to right. A time series shown for the advance would be oriented more naturally.

Figures 1.8 and 1.9 show the result of directly plotting army size and air temperature. In the first figure we use the (reversed) longitude for the horizontal dimension, and in the second figure we show the time. Not only can the drop in

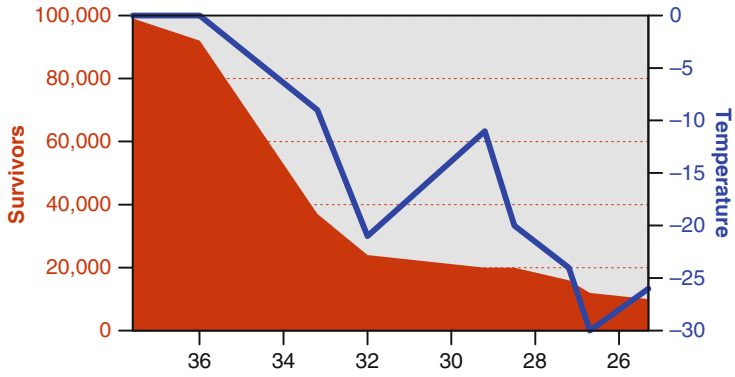


Fig. 1.8 Napoleon's retreat. An area element shows the size of the main component of the army (survivors) and a line shows the temperature. These have different *y* dimensions and are color coded to match the corresponding axis. The *x* dimension indicates the position of the army as a longitude. The *x* dimension has been reversed so that time runs *left* to *right*

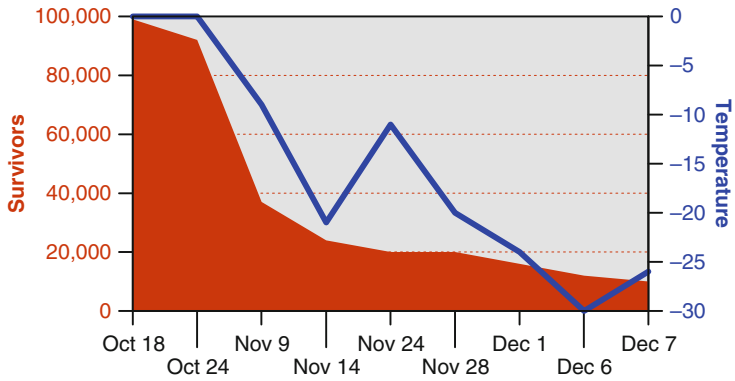


Fig. 1.9 Napoleon's retreat. An area element shows the size of the main component of the army (survivors) and a line shows the temperature. These have different *y* dimensions and are color coded to match the corresponding axis. The *x* dimension indicates the date

troop numbers and temperature be seen, but by comparing the two figures we see that there is not much distortion in the way that time and distance are conflated in Minard's chart.

At an overview level, at least for the main component of the army in the retreat direction, the two charts can be thought of as the same. These additional views give us confidence in Minard's technique, so that we can make assumptions about the horizontal dimension as a time dimension and know we are not being misled.

1.4.2 Summary

Minard had a purpose with his chart; he was not a casual historian looking at some interesting data who decided to draw a beautiful chart to illustrate it. Rather, he disliked the cult of personality surrounding Napoleon and wanted to show people just how large a disaster the attempted conquest of Russia was. Minard started with a goal; the graphic he created was motivated and purposeful. Since the goal was to show that Napoleon's campaign was a failure, he needed to answer the question of what made it a failure, and the chart answers that question. To answer the question, Minard needed data, and in the annotations for the chart he describes the sources he chased down to find the data he needed.[6] This shows an excellent strategy for designing a visualization – start with a goal, define the questions that need answering to support that goal, and gather data for visualizations that answer those questions. This approach will be used as a foundational approach in Chap. 3. This is not to say that the only goal of visualization is to support a specific conclusion; Minard's goal is to explain and explore the known result, but often visualization will be used to discover those results.

Minard, like Playfair, wants to display time × quantity on his chart. In this case, the quantity is a measure of size – the number of survivors in the army. The horizontal dimension is doing double duty as both *time* and *longitude*, but the vertical dimension is also being used to show position,[7] so the natural mapping used by Playfair is not possible. With the positional dimensions used up, displaying the army movement by a path, Minard uses a *size aesthetic to show a size variable* – the numbers of survivors are shown using the width of the path. This, as is seen in Chap. 7, is a natural, interpretable, and highly effective mapping.

Minard does not give many guides. Figure 1.7 adds a legend for the direction of the army, but is it really necessary? The size of the army is indicated by labels in Minard's figure, but once we know that it starts at 480,000 and ends at 10,000, the only reason to include actual values or a legend is if the original data are unavailable to us and we want to read values directly from the chart. In almost all circumstances, it is better simply to include a tabular form of the data if reading values are required and free the chart from clutter. Minard wisely avoids adding gridlines for latitude and longitude or intrusive legends, letting the data stand out clearly.

Again like Playfair, Minard uses horizontal alignment to link two charts together, but whereas Playfair simply juxtaposes them, Minard adds lines explicitly showing the linking, and his linking is more complex, not just showing how two similar charts are related through time, but taking two charts with different quantities and

[6] In Fig. 1.7, the French text toward the top reads, in part, "The information used to draw up the chart has been taken from the works of M. M. Thiers, of Segur, of Fezensac, of Chambray, and the unpublished diary of Jacob, pharmacist of the army since October 28th."

[7] It would be much too fortunate to discover that the size of the army was correlated with the *latitude*!

different appearances and linking them together. This is a much larger step toward *interactive linking* of charts and is a major innovation in charting.

A final detail worth noting is providing *guides to show context*. Minard adds geographic detail to his main chart that allows us to see towns, rivers, and battles that help the viewer to understand patterns and draw conclusions. The resulting chart is complex, but compelling. It fulfills the designer's goal but does so in a way that also invites us to explore other hypotheses, and it allows us to draw other conclusions. Not only do we see the main effect – many left for war; few returned – but the chart leads us to ask other questions, such as:

- How did low temperatures affect mortality?
- How dangerous were river crossings?
- Is the combination of the above two factors particularly serious?
- Relatively speaking, were the battles a major cause of death?

This chart is based on a fortunate coincidence, but it does not rest on its laurels. A good story has subplots as well as plots, and a good visualization answers more than just one question. The genius of Minard's chart is that it delivers on the major goal while suggesting answers to other questions. And it looks good doing it.

1.5 Comic Books

Comics are a widespread medium for telling a narrative. They are used for presentation, as in the example shown in Fig. 1.10. They are also used as collaborative exploratory tools for the creation of movies and stage art, where they are typically called "storyboards." Comics are a highly successful form of presenting stories in visual form and are intuitive, attractive, and easy to use. Kaplan [65] states the following:

> Comic books reigned supreme among the new mass-market media of the 1940s. For sheer entertainment value, they had almost every other media outlet beat. Unlike radio, comic books could actually show the action being described by the narrator. Unlike theater, comic books could take you to other worlds, other dimensions. Unlike movies, comic books were always in vibrant, living color, and budgetary restrictions weren't an issue given that the special effects department consisted of a penciler, an inker, and a colorist who were limited only by their imaginations.

Comics are often thought of as an American phenomenon, but they are pervasive in many cultures, with the popularity of manga in Japanese culture being a prime example. Although the exact definition of a comic is often debated by experts, the essentials of a comic include the following elements:

- A series of panels that display the story at certain points in time
- Text boxes and speech/thought balloons that are used to convey narrative information

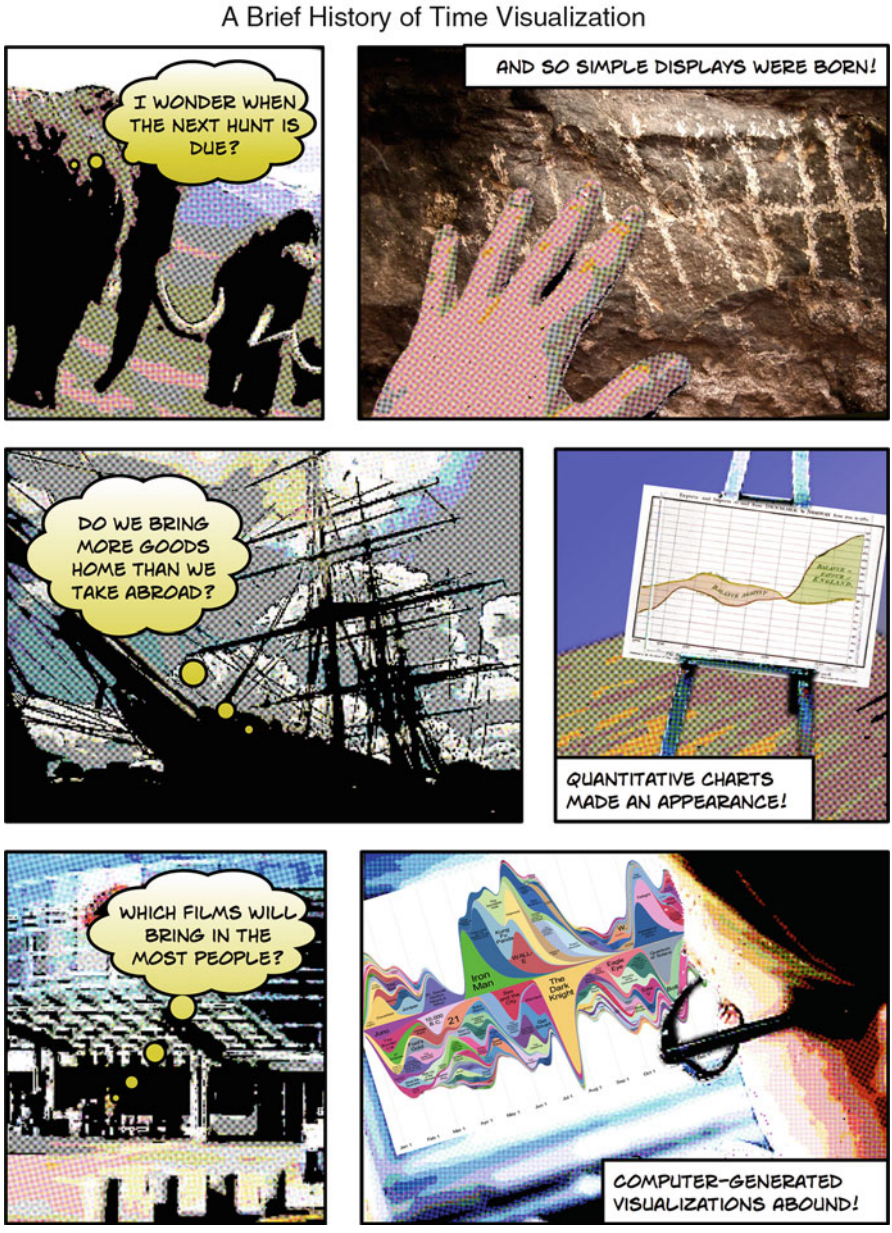


Fig. 1.10 A whimsical look at the evolution of charts over the past few thousand years.

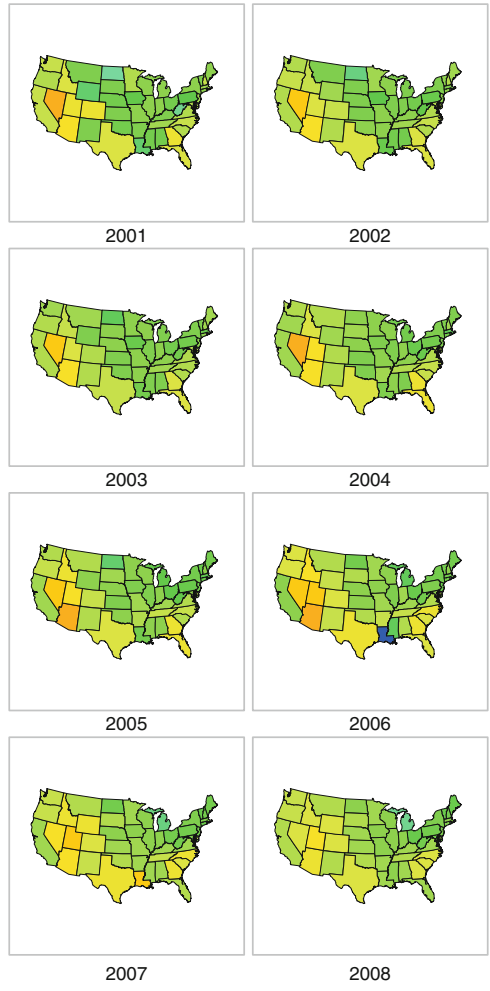


Fig. 1.11 A paneled (or faceted) layout showing the percentage changes in populations of states in the USA in the period 2001–2008. The layout used for the panels is a wrapped layout, similar to the method used in comics. The figure displays only the 48 contiguous states

- Panel layout, zip ribbons (motion lines), and other annotations that are used to convey the flow of time

Figure 1.11 shows an automatically generated paneled figure using the same layout metaphors, but rather than being narrative focused, it is data focused. Each panel shows the state of the data at a given point in time, namely a year. The data are taken from the US Census Bureau and show the changes in population relative to the previous year. Comparing this data display to the comic of Fig. 1.10, note the following points:

- The panels do not vary in size. Because we are showing data, we want the viewer to be able to compare quantities visually, and so fixing the panel sizes is important. The basic layout of the panels is the same – in reading order.[8]
- Panel titles are shown outside the frame, rather than inside. This is not universal, but many charts have multiple labels inside, and so text that titles a panel is commonly placed outside it. However, text that is attached to data elements, which in comics is typically speech and thought, is placed in the frame. In fact, the use of lines, bubble extensions, or series of circles (as shown in Fig. 1.10) has been copied by the charting community, as shown in Fig. 1.12.
- In comics, zip ribbons (also called "swoosh lines" or "motion lines") are used to show which objects are moving. In the data visualization world, it is also important to highlight what has changed since the previous time slice. In Fig. 1.11 the data displayed are not the raw data of population counts, but the changes between counts. This may seem like an unnecessary step – you can just compare a panel with the previous one, but that comparison is not an easy one to make, so directly displaying the changes between panels is a valuable technique.

1.5.1 Summary

Comics provide a way of moving time from a temporal dimension into a spatial dimension. A comic uses *faceting*, or paneling, by which slices of time are placed in a sequence on a page. In this way a comic allows comparison between figures, which allows us to see how patterns and stories evolve. Our perceptual system is much better at flicking between two figures to compare them than it is at keeping items in short-term visual memory and remembering how they used to appear. Ware [124], discussing how we process visual objects, notes that only one to three visual objects make their way through our cognition processes into visual working memory. Therefore, we cannot expect to make many comparisons if we are looking at a temporal flow of information. It is much more efficient to allow our eyes to flick between objects on the same page – making comparisons and seeing how things evolve. Comics and storyboarding take complex objects and allow us to see how they change over time. The same technique is no less useful for visualization of time, as we examine in Chap. 6.

[8]Not all cultures read in the same direction! If you buy a manga in Japan, you will find that the direction of reading is reversed; from a Western viewpoint, you open the last page and read backward. When translated for foreign readers, mangas are often reversed by flipping the horizontal dimension, to allow Westerners to read them in their natural direction. One unfortunate effect of this is that everyone then becomes left-handed, which can be disconcerting, especially when studying sword fights in classic manga such as [69]. However, despite this difference in reading direction, charts with time in them are still generally displayed in left-to-right orientation.

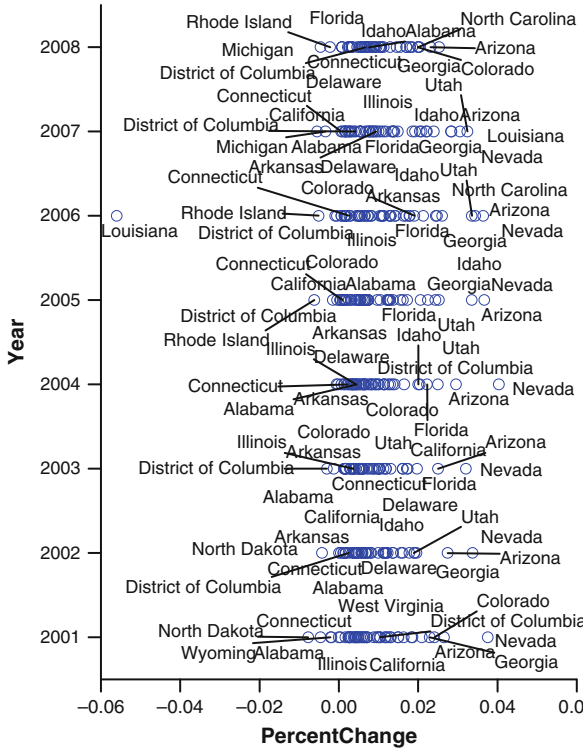


Fig. 1.12 US state population changes, 2001–2008. The US state data shown as a scatterplot of year against change percentage. The states are labeled, but since there are too many labels to draw, a set of modifications has been made in an attempt to alleviate the problem. First, the labels are moved around to try and place as many nonoverlapping labels as possible. Second, overlapping labels have been removed to prevent overdrawing. Third, connecting lines have been drawn that connect labels to points when they have been moved far away from their original location. The result is by no means ideal, and a manual process of label deletion or addition would make the plot far superior, but as a first attempt, it is at least better than simply plotting all 384 overlapping labels

A second contribution is the use of labels and captions. Not only do comics have a large amount of text, but the text is of different types and may need to be attached to different elements clearly. The use of *aesthetic properties on the text boxes* allows us to understand the different sorts of data that are being displayed. In a comic, we can tell who is generating the text and whether they are thinking, speaking, or shouting by the way the text and surrounding text box are formatted. The same techniques allow us to present textual data and give additional information on its meaning, as we show in Chap. 7.

Comics provide additional hints as to what is changing, and how much, by the use of zip ribbons and similar modifications to elements. For data visualization, we can take this technique in a couple of directions. We can add aesthetics that show the change, or we can directly change the data being displayed to show changes,

rather than the raw data. Figure 1.11 tends more toward the latter, as it replaces the absolute population levels with change values. We could instead have shown the raw populations and then maybe added zip lines, a background blur, or simply an aesthetic on the boundaries of the state shapes if we simply wanted to annotate the base data.

1.6 Further Exploration

The Web has a fair amount of interesting information at a high level on the history of visualization; a good place to start is with the site Milestones in the History of Thematic Cartography, Statistical Graphics, and Data Visualization [46], which can be found at `http://datavis.ca/milestones` (it was previously at `http://www.math.yorku.ca/SCS/Gallery/milestone`).

Tufte's *The Visual Display of Quantitative Information* [111] is a classic, and of his books, it provides the most of interest from a historical angle. All his books are worth reading, but this one gives the best historical perspective both for time and nontemporal visualization.

Wainer's *Graphic Discovery: A Trout in the Milk and Other Visual Adventures* [122] contains a wealth of information on Playfair, both from a visualization point of view and also from a pure "human interest" angle.

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*,[1] 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.

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

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_2,

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.[2]

[2]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 *a*xes as guides and represented by a *p*oint element. A bar chart of counts consists of a single variable representing categories, placed in a *2*-D rectangular coordinate system with *a*xes as guides and represented by an *i*nterval element with a *c*ount 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 *e*lement 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.

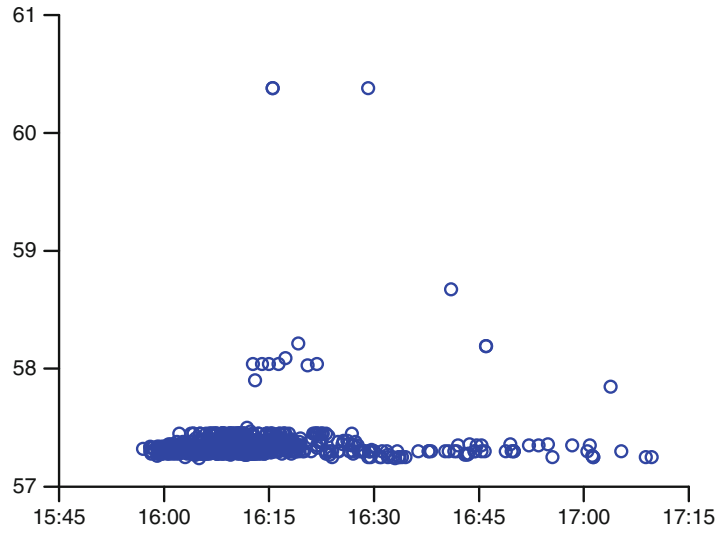
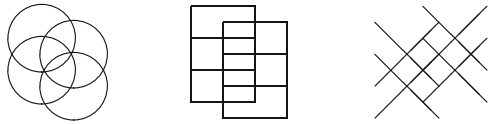


Fig. 2.1 Stock trades: price by time. A scatterplot: two variables in a 2-D *coordinate* system with axes; each row of the data is represented by a *point* The data form a subset of trade data for a single stock, with each point representing the time of a trade and the price at which it was traded

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

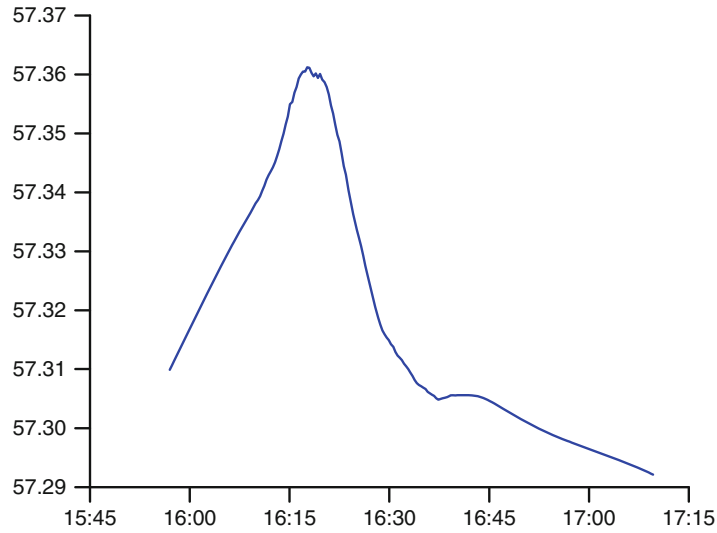


Fig. 2.2 Stock trades: price by time. Line chart: two variables in a 2-D coordinate system with axes; a single *line* represents all the data. A smooth statistic (Sect. 2.3) has been applied to the data. The data are the same trade data of the previous figure

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

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

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

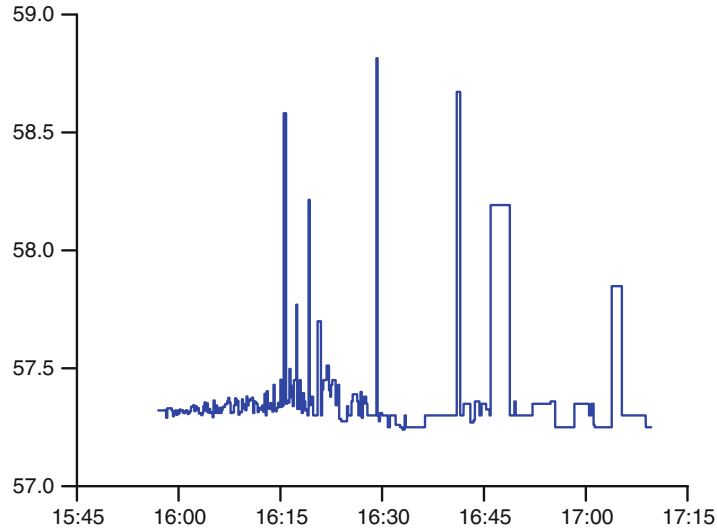


Fig. 2.3 Stock trades: price by time. Step representation of a line chart. This is the same chart as in Fig. 2.2, except that we have used a step function on the data so it does not interpolate smoothly between values, but instead steps abruptly

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 *s*tacked, 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

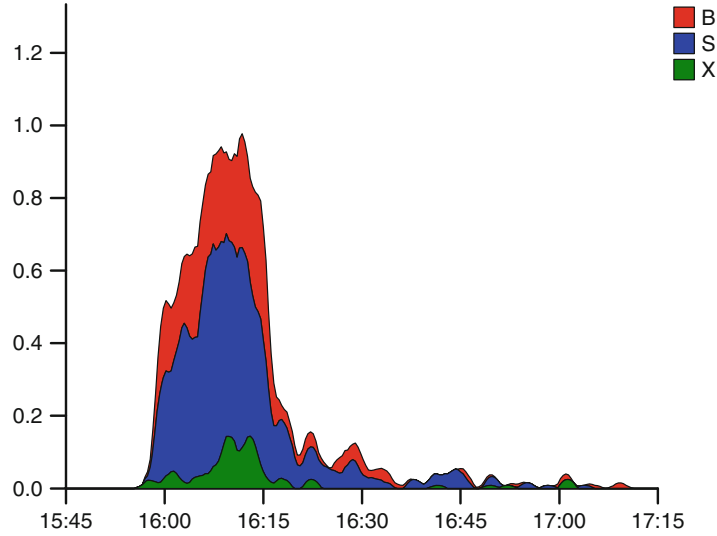


Fig. 2.4 Stock trades: volume by time. An area chart: two variables in a 2-D coordinate system with axes; an area element is displayed for each group in the data. The groups are defined by the *TradeType* variable, which indicates whether the trade was a buy, sell, or cross-trade. For each group, an 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 $startingtime \times endingtime$ 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 $y = velocity$.

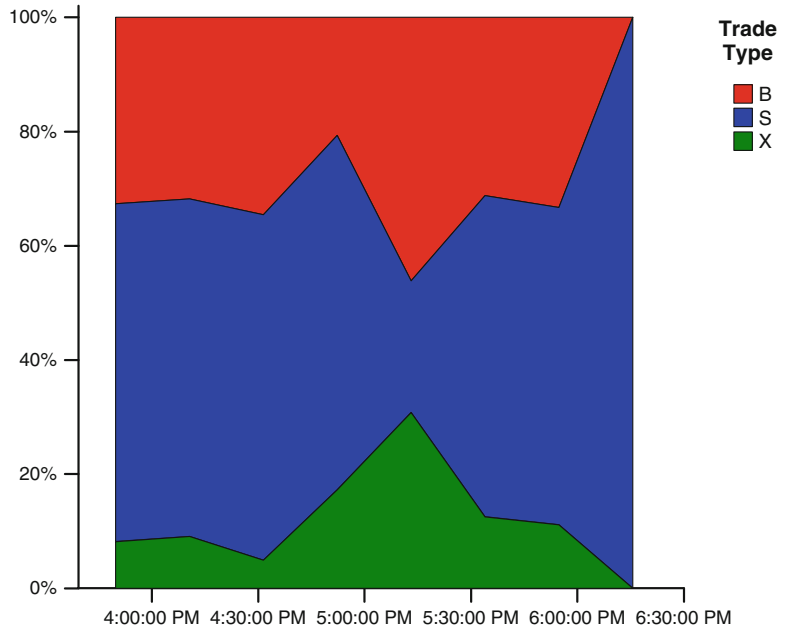


Fig. 2.5 Stock trades: ratios of types by time. A modified version of Fig. 2.4 in which the density statistic has been replaced by a statistic that first bins the data using the horizontal (time) dimensions and then calculates the percentage of each group within each bin. The result shows the changing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

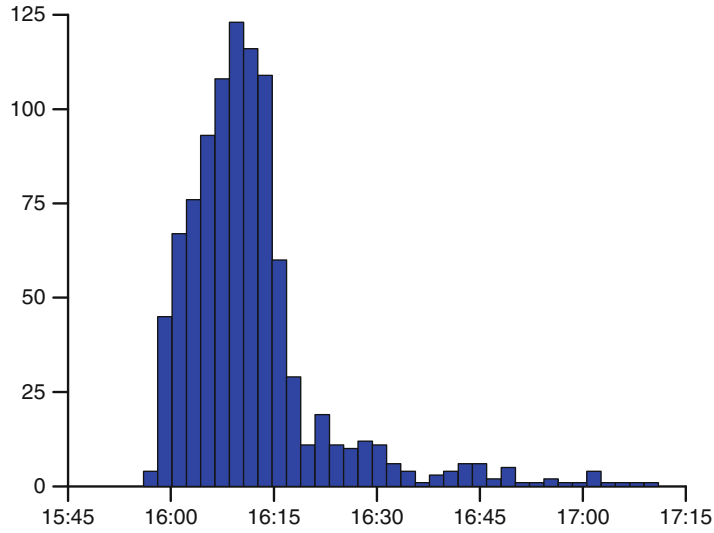


Fig. 2.6 Histogram of trade times: one data variable in a 2-D coordinate system. The second dimension is generated from the first by a pair of statistics. The first statistic *b*ins the *x* values into disjoint bins, and the second statistic *c*ounts the number of rows that fall in each bin. This gives 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 *c*olor *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.[3]

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

[3]In Sect. 2.3 we will explain a little more about statistics. In particular we will deal with the use of *w*eight variables, the use of which is the best way to describe this data set for most purposes.

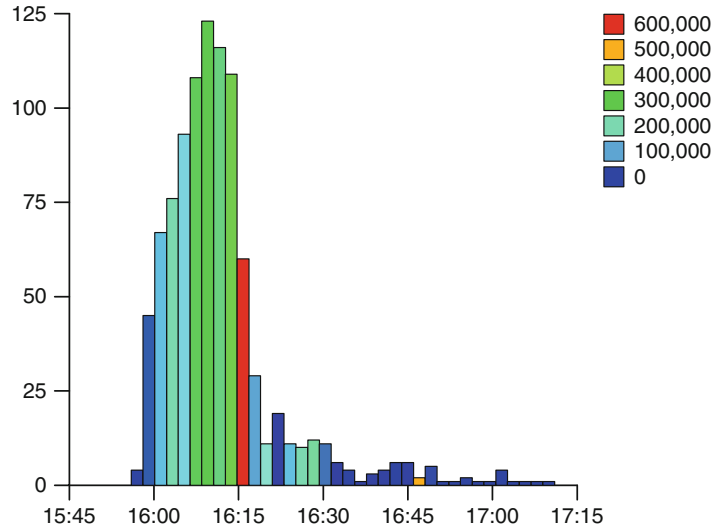


Fig. 2.7 Histogram of trade times. This figure is similar to Fig. 2.6, but we have colored each *bar* by the sum of trade volumes in each *bar*. We can see in this figure that, although most trades took place between 4:10 p.m. and 4:15 p.m. (local time) the time period just after this period saw more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

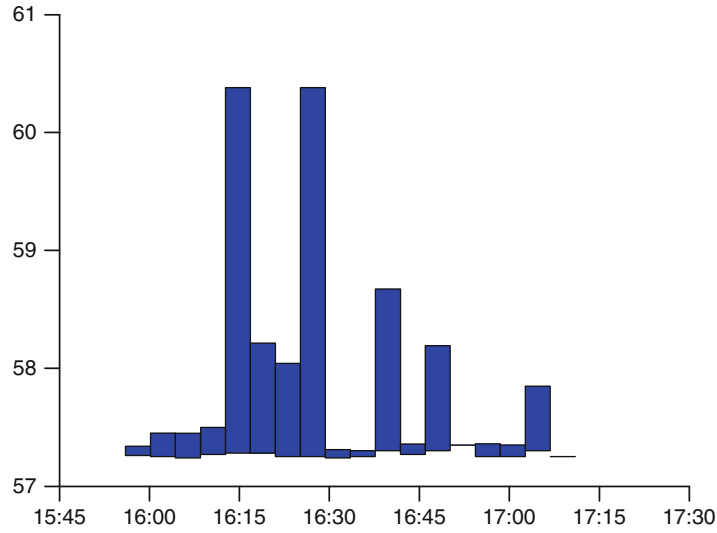


Fig. 2.8 Range plot of trade times: two variables in a 2-D coordinate system with two chained statistics. The first statistic bins the x values into disjoint bins and the second statistic calculates the range of y values in each bin. This gives a representation of the range of trade prices over time 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.[4]

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

[4]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.

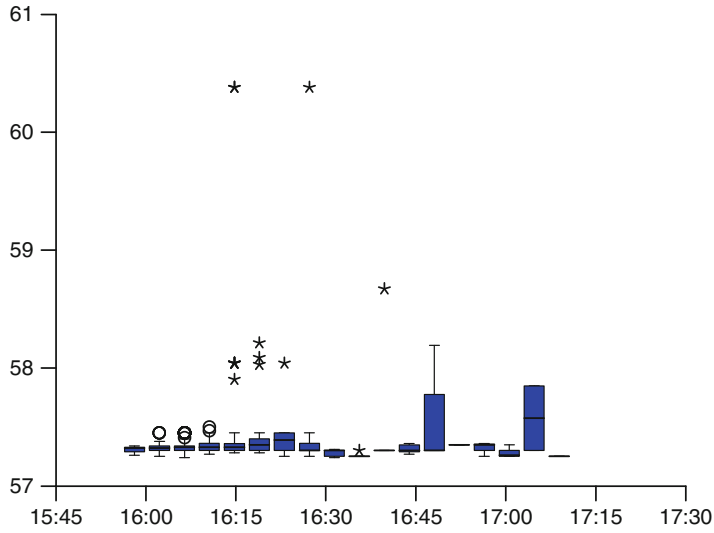


Fig. 2.9 Stock trades: price by time. Boxplot: two variables in a 2-D coordinate system with two chained statistics. The first statistic bins the x values into disjoint bins and the second statistic calculates the *Tukey statistics* of y values in each bin The data form a subset of trade data for a single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

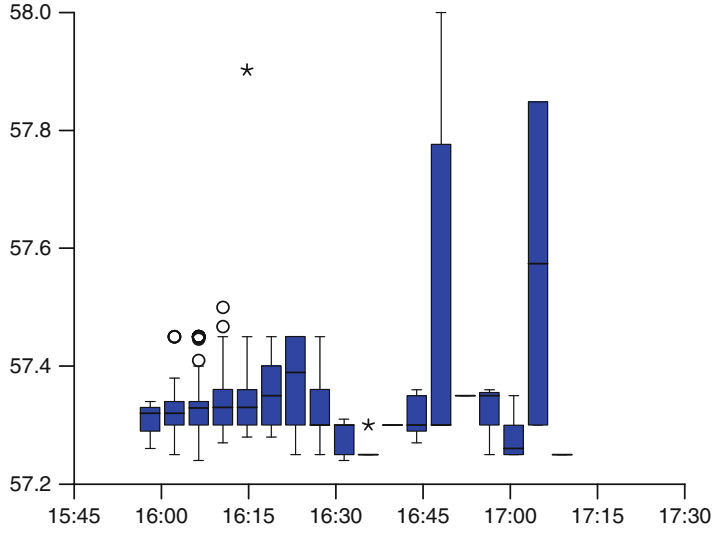


Fig. 2.10 Boxplot: the same graph as in Fig. 2.9, but restricting the y dimension to show a smaller range

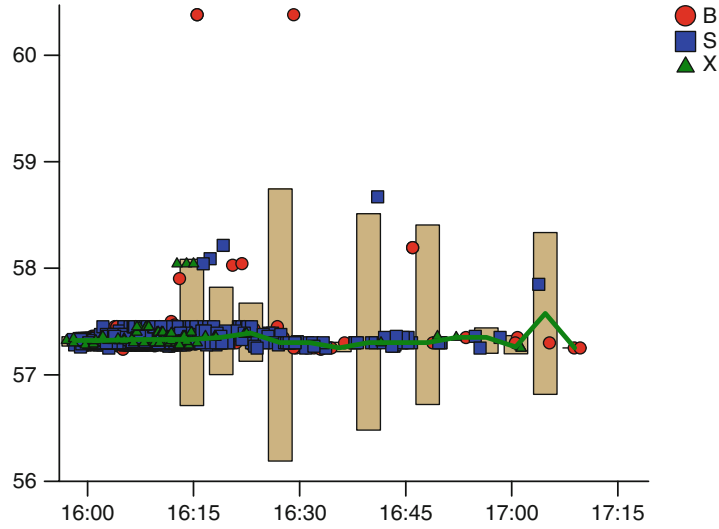


Fig. 2.11 Combination of three elements displaying trade price by time in two dimensions. A point element showing individual sales, an interval element showing, for binned values, the 95% confidence interval for those values, and a line element showing the median values in the same bins

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

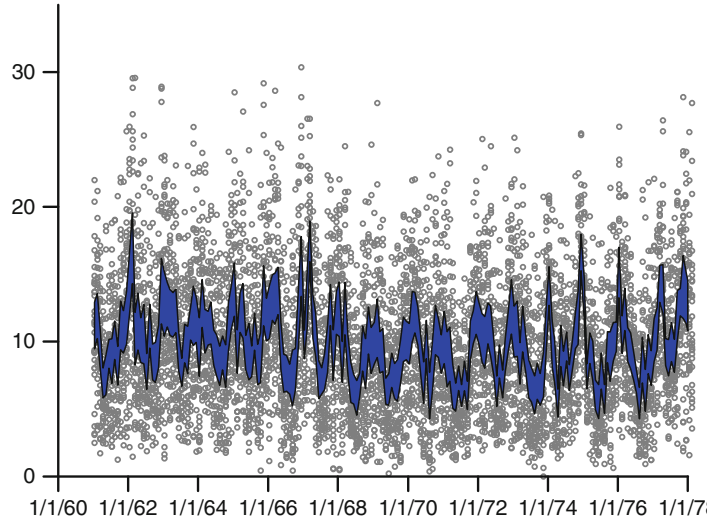


Fig. 2.12 Wind speeds measured daily in Dublin, Ireland for the years 1961–1978. On *top* of the raw data is superimposed an area element showing a 95% confidence interval for the mean wind speed in each month

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[5] 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

[5]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.

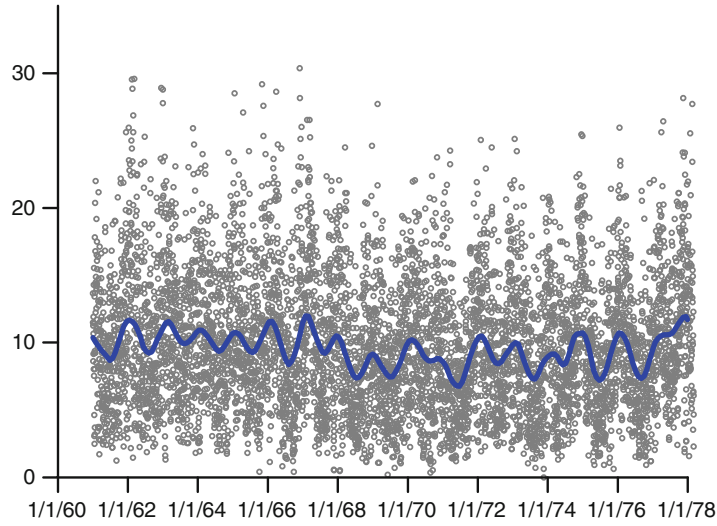


Fig. 2.13 Wind speeds measured daily in Dublin, Ireland for the years 1961–1978. The figure shows both the raw data and a loess smooth of the data. The loess smooth uses a window size of 1 year and an Epanechnikov kernel

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.[6]

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

[6]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.

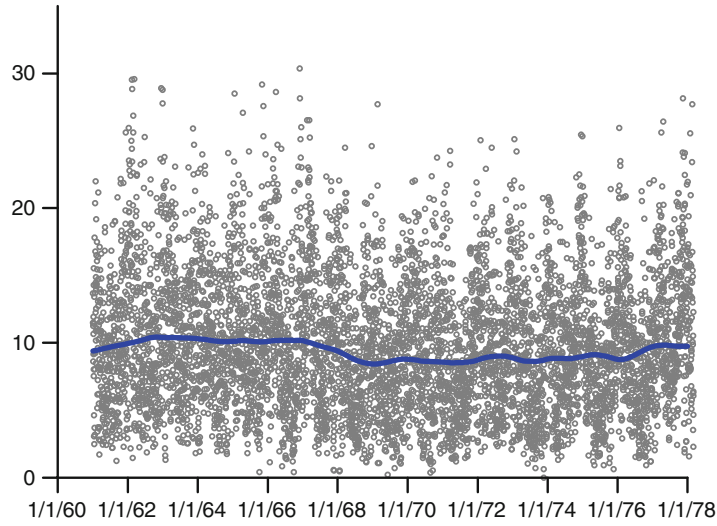


Fig. 2.14 Wind speeds measured daily in Dublin, Ireland for the years 1961–1978. The figure is the same as Fig. 2.13, except that the loess smooth uses a window size of 3 years, which smooths out the seasonal trends

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

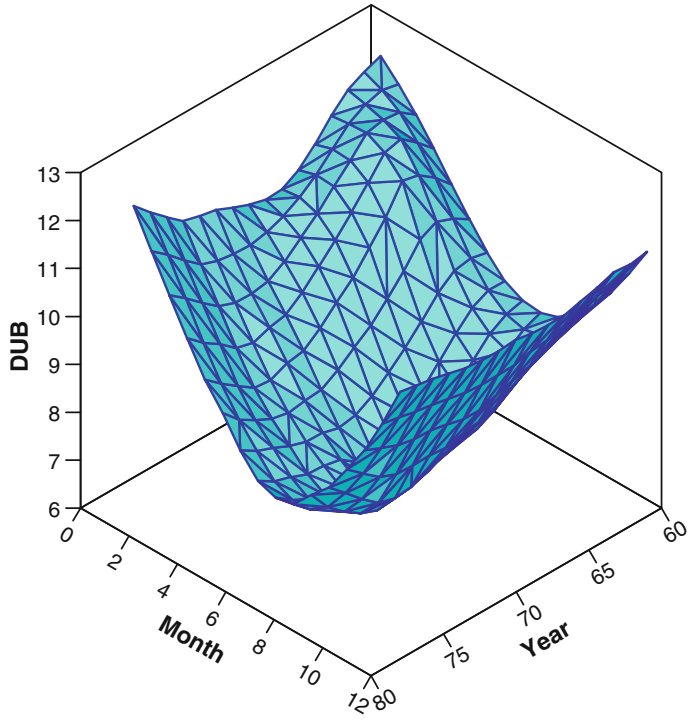


Fig. 2.15 Wind speeds measured daily in Dublin, Ireland, by year and month. The *date* variable has been split into two dimensions, one for year and one for month. A surface for the loess smooth has been plotted

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.

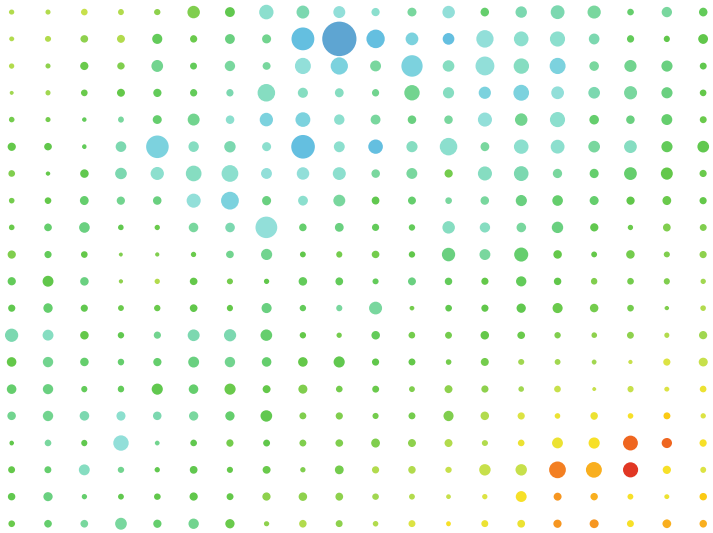


Fig. 2.16 A self-organizing map of wind data using wind speed data for six cities as the six variables used to create the plot. The size of the *point* at a grid coordinate indicates how many rows are mapped to that location; the *color* represents the average wind speed for that grid location

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.

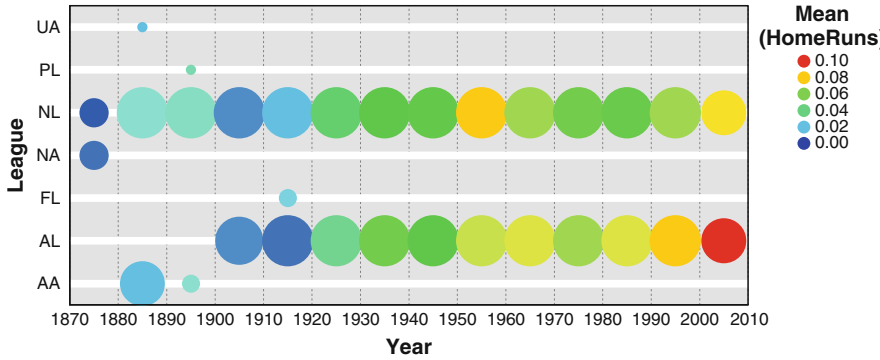


Fig. 2.17 Baseball players' average number of home runs per game, binned into decades and split vertically by league. The color of the *points* encodes the average number of home runs per player per game

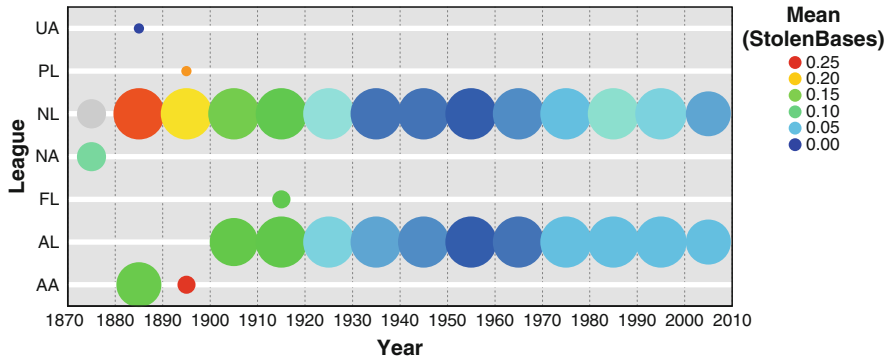


Fig. 2.18 Baseball players' average number of stolen bases per game, binned into decades and split vertically by league. The color of the *points* encodes the average number of bases stolen per player per game

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.[7] 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`. 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

[7] `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.

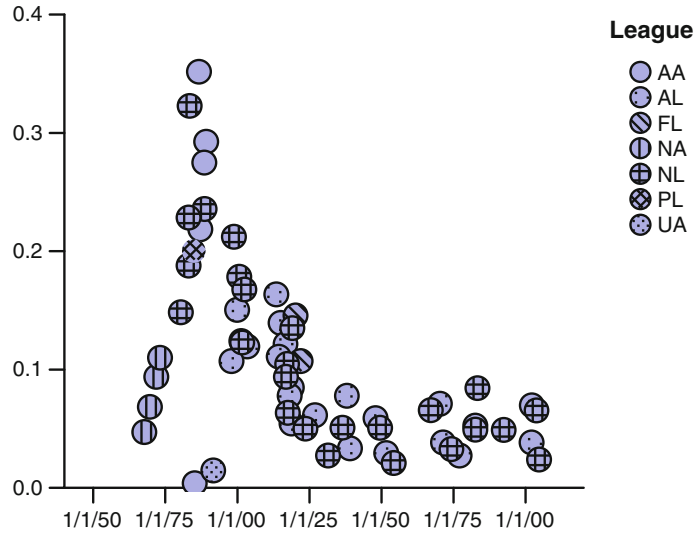


Fig. 2.19 Baseball players' average number of stolen bases per game. This figure shows the same information as Fig. 2.18, namely, how many bases are stolen per player per game, aggregated into decades and split by league using a *p*attern aesthetic. The *points* have been jittered to reduce the effect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

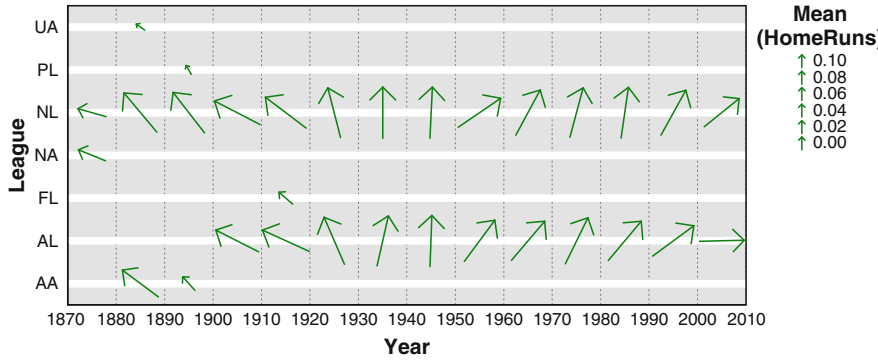


Fig. 2.20 Baseball players' average number of home runs per game, binned into decades and split vertically by league. The angle of the *arrows* encodes the average number of home runs per player per game

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.

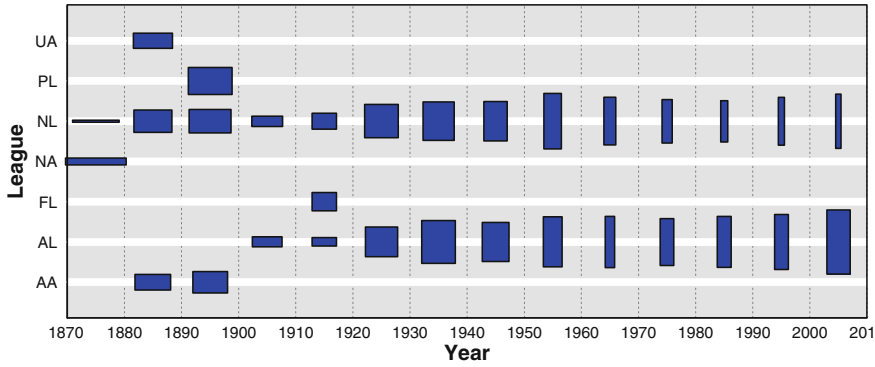


Fig. 2.21 Baseball players' performance by decade by league. The width of the *squares* encodes the square root of the average number of runs scored per player per game, and the height of the *squares* encodes the square root of the average number of home runs scored per player per game

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.

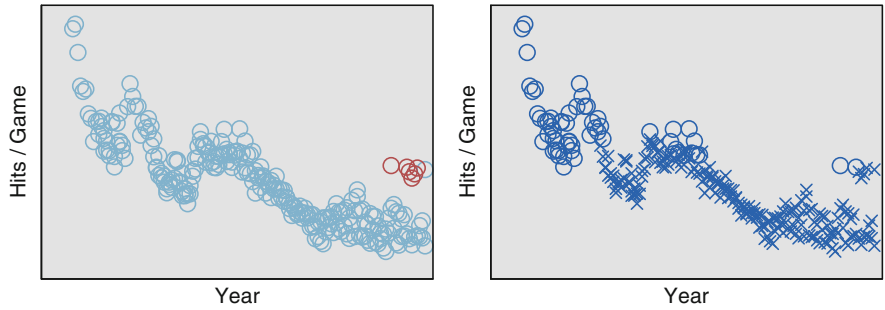


Fig. 2.22 Separate aesthetics used on a simple scatterplot of hits/game against year. The data are statistics on average player performance aggregated by league and by year. On the *left* we highlight cases where players average more than one home run every ten games using a color aesthetic where *red* 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

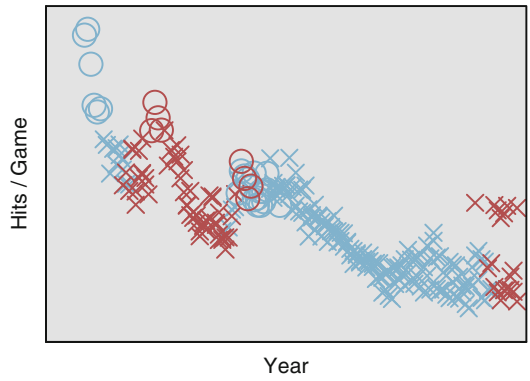


Fig. 2.23 Baseball player performance: hits/game against year. The data are statistics on average player performance aggregated by league and by year. *Red symbols* represent seasons where players were hit by a pitch more often than one game in 40. *Circles* indicate seasons where players averaged less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

higher hitting average.[8] 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

[8]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.

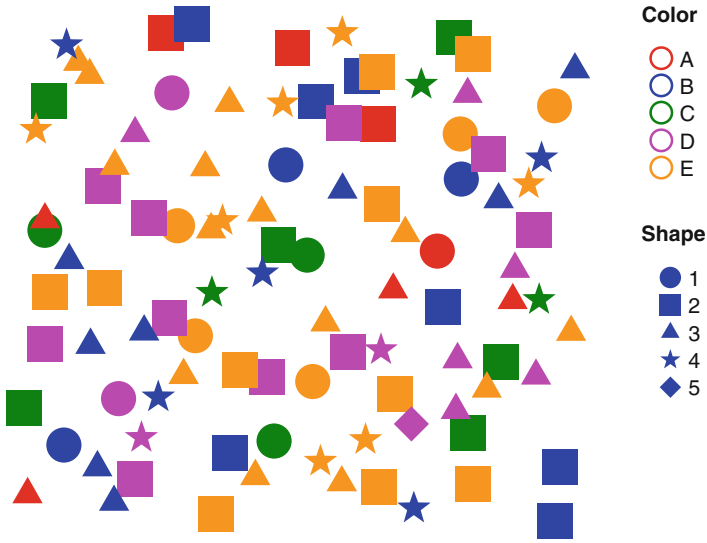
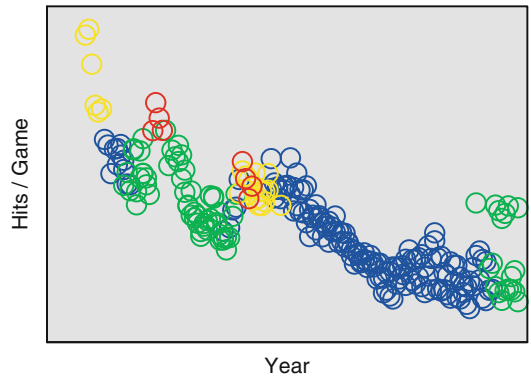


Fig. 2.24 Synthetic data set of 100 points with color and shape aesthetics randomly assigned

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.

*C*oordinates of a graph tell us how positional variables are to be used in the chart. *F*aceting, 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:

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

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 (ϕ) 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

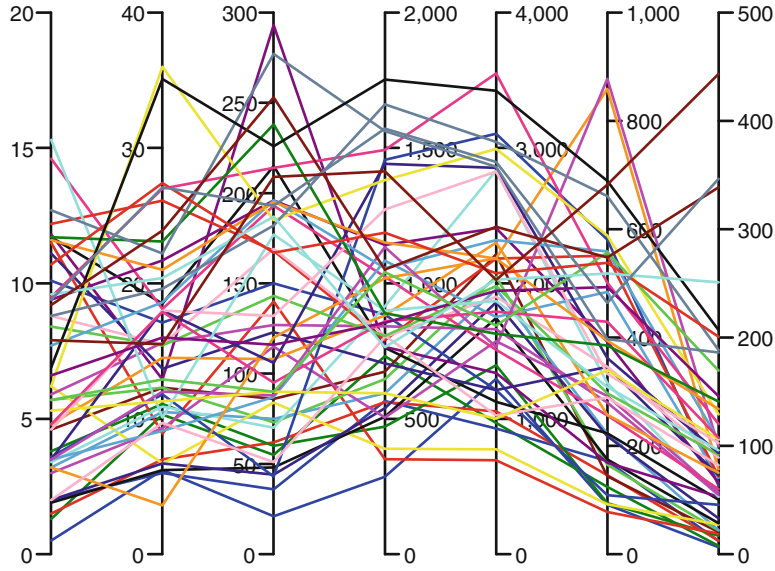


Fig. 2.26 A parallel axis plot showing crime statistics for US states. The variables being plotted on each axis represent rates of occurrence of several categories of crime. *left to right*:

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

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,[9] 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?

[9]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

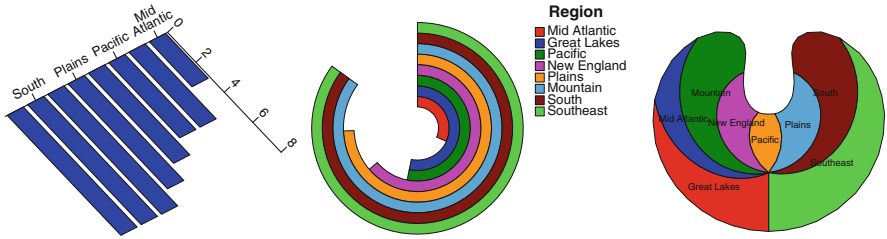


Fig. 2.27 Three syntactically correct, but not terribly useful, coordinate chains. The first consists of a rotation transform followed by a nonproportional scaling of both dimensions. The second takes a simple bar chart and, instead of simply applying a polar transform (which would produce an at least *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 ϕ 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 ϕ 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

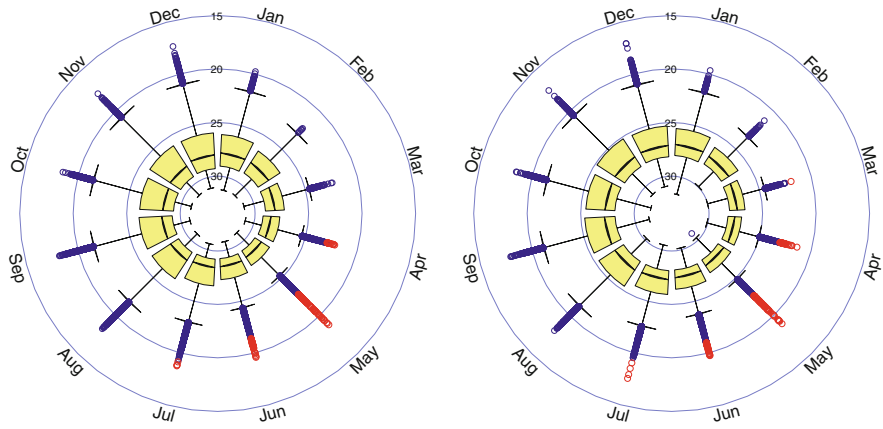


Fig. 2.28 El Niño temperature data. These data consist of 178,080 observations of sea and air temperatures over the period March 1980–June 1998 for a number of geographic locations. In these figures we show the sea temperature (*left*) and air temperatures (*right*) for all observations, conditioning 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*

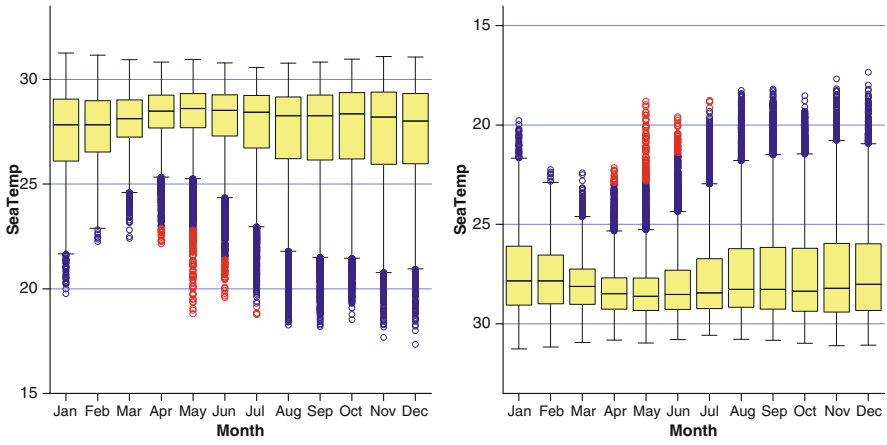


Fig. 2.29 El Niño sea and air temperatures. Two steps in the coordinate chain that finishes with Fig. 2.28. On the *left* is the original figure; on the *right* we have flipped the y dimension. To create Fig. 2.28, we then apply a polar transformation

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

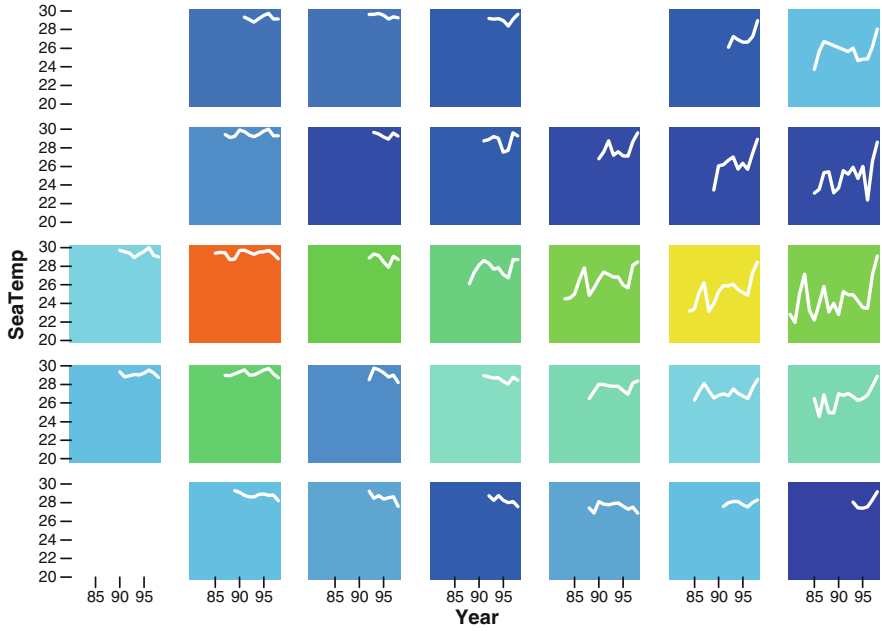


Fig. 2.30 El Niño: spatial and temporal data. The faceting dimensions show the spatial locations of the buoys that collect the data. Within each facet cell is a line chart showing mean sea temperature by year. The color of the cell indicates how many observations were recorded in that cell

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 7×5 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:

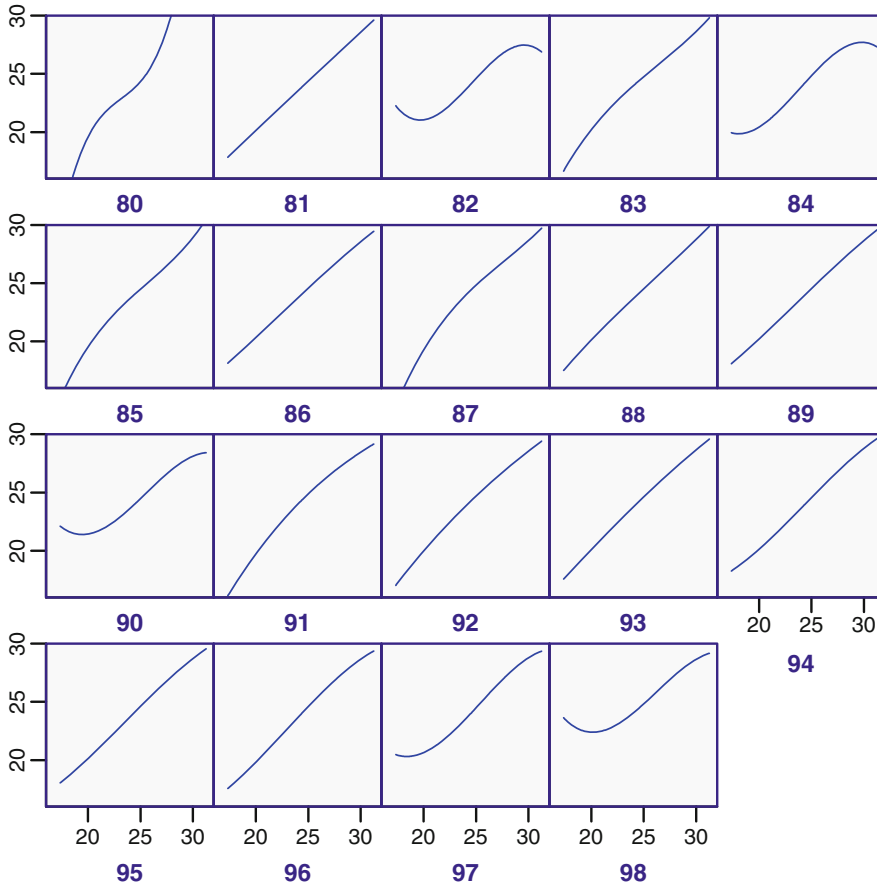


Fig. 2.31 El Niño data. A line chart showing a cubic smooth on the relationship between *AirTemp* and *SeaTemp*, faceted by year. Clear differences in the relationships can be seen. For reference, strong El Niño effects were observed in 1982–1983 and 1997–1998

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}
```

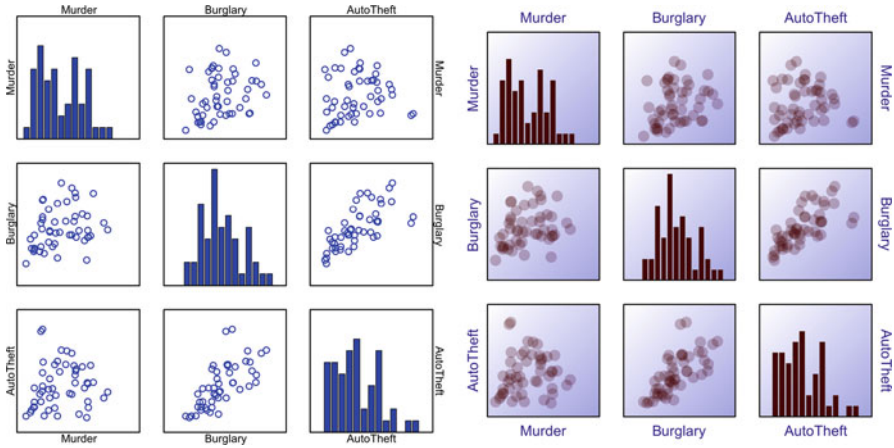


Fig. 2.33 Crime statistics for US states displayed as a *scatterplot matrix* or *SPLOM*. The plots on the *left* use the default styles employed in this book; those on the *right* use more ornate styles

```
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,[10] 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.

[10] 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 *a*esthetic 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 *T*he 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.

Chapter 3
Designing Visualizations

Study without reflection is a waste of time; reflection without study is dangerous.

— Confucius, *Analects* (551–479 BCE)

3.1 Guiding Principles

Having recently joined IBM, I attended a series of introductory sessions for new employees. While there were a number of interesting points (including the fact that IBM had a clear antidiscrimination policy back in the early 1950s), I found that in particular one of the company's values struck a resonating chord: "Innovation that Matters."

With a little modification, that motto forms a good subtitle for this chapter, the goal of which is to help people design *visualizations that matter* – visualizations that convey information that allow people to find and solve problems, discover relationships, and make better decisions. Although focusing on time data, much of the material in this chapter will be generally useful. Indeed, the main approach used is based on a technique that comes from software engineering, the GQM approach.

The goal of this book is to provide a guide to the creation of useful visualizations of data, specifically data with a time component. The meaning of "a useful visualization" is a visualization that can be used for one of the following two purposes:

1. To learn a feature of the data that is being represented.
2. To display some known feature of the data.

A visualization fulfilling the first goal is often termed an *exploratory visualization*; one that is aimed at the second goal is often termed a *presentation graphic*. An important point to note is that the difference between the two is not one of content, but of *goal*. A scatterplot might be thought of as an exploratory chart and a pie chart

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_3,

a presentation chart, but only because the former is more often used for exploration and the latter for presentation. It is not an intrinsic feature of the chart. For this reason it is not useful to state rules that relegate charts to one division or another. A scatterplot can be used to drive home an important point, and a pie chart can be used to explore data. This chapter looks at choosing charts and designing new graphics from the viewpoint of achieving goals – helping us to design visualizations that lead to revelations that convey knowledge that allow people to make decisions with more ease.

Although the focus of this chapter is on utility, that is not to say that making charts attractive is a bad idea. Most often, an ugly chart is not as useful as a beautiful chart – the two go hand in hand. It is even (and this notion might be heretical to some visualization practitioners) acceptable to make a nonoptimal choice in a chart from a utility point of view so as to make it more presentable. This compromise occurs many times in this book; charts have been compressed or have had their aspect ratio changed so as to fit available space; white space and borders have been applied to make a more pleasing composition, which shrinks the proportion of the chart used to show the data; choices of which aesthetic to use for a variable have been made to make charts look better; more visually complex charts have been chosen (using the criteria of Chap. 11 to determine complexity) than were needed, and so on. Guidelines are more helpful than rules.

3.1.1 The GQM Methodology

GQM was invented in the early 1990s by Victor Basili. It is a software engineering methodology that organizations can use to be effective in their use of (software) metrics. The philosophy of the approach is stated in [3] as follows:

> The Goal Question Metric (GQM) approach is based upon the assumption that for an organization to measure in a purposeful way it must first specify the goals for itself and its projects, then it must trace those goals to the data that are intended to define those goals operationally, and finally provide a framework for interpreting the data with respect to the stated goals. Thus it is important to make clear, at least in general terms, what informational needs the organization has, so that these needs for information can be quantified whenever possible, and the quantified information can be analyzed as to whether or not the goals are achieved.

Collecting data to serve a purpose, rather than just doing so because it seems like a good idea, is the aim of GQM. In a similar vein, we would like our charts to be useful rather than just filling space and looking pretty. Although the GQM approach can be used directly with visualization, in most cases when visualization is considered the data have already been collected, and so our usual objective is to create useful charts with the data we have. In the rest of this chapter, we therefore concentrate on the first couple of steps – *goals* and *questions*. Section 3.2 discusses the major goals that can be achieved with visualization, and Sect. 3.3 describes

the questions that can be answered with specific graphical solutions. The nuts and bolts of the data that go into a chart and how best to portray those data using the various grammatical components is a core subject dealt with throughout this book. Therefore, the last step, which is described in Sect. 3.4, is only a brief discussion of how mappings from data to chart components fit within the design approach described here. Details on the mappings to aesthetics, coordinates, facets, and so on can be found in subsequent chapters.

3.2 Goals

When establishing a goal for a visualization or a set of visualizations, a number of factors should be considered:[1]

Object Decide what is being studied and presented. Without a clear idea of what the field of study is, you are in danger of leaving out important details or of adding in useless details. Also, decisions need to be made on the relative importance of different data. For example, if the object is to study *network traffic*, information on the physical hardware is of secondary importance to data on the packets being sent. On the other hand, if the object is *network reliability*, then the order of importance may well be reversed, with the hardware being more important.

Purpose Why are visualizations needed? What sort of action do we expect people to take based on their observations?

User Knowing not just the user but how the user will use the visualizations is of prime importance. Is he monitoring a stream of information or exploring a system at leisure? Is she knowledgeable about visualization in general? Does he have particular charts he currently uses? How much does she know about the data she wishes to graph?

There is one fundamental decision that needs to be made when establishing the goal – is the user going to be working in an *immersive* or a *reflective* mode? In an *immersive* mode, people are actively engaged with the data, making decisions, acting and viewing as part of a flow of work. An example would be a system for fraud analysis; the user is involved within the system, using visualization as one tool to understand the data, make many quick decisions, and act on information obtained. By contrast, a *reflective* system is one where the user expects to examine the graphs at more leisure, to think of possible hypotheses about the data, and then explore the visualization to confirm them. The user "stands outside" the data, reflects on their features, and draws conclusions. In the immersive system, the user "lives within"

[1]This list is adapted from a subset of the standard GQM template descriptions.

the data, checking and validating known features, and searching for answers rather than drawing conclusions. In some ways this distinction is tied to the difference between presentation and exploratory graphics – the goals are different: to see and act, or to interact and learn.

3.2.1 Presenting What Is Important

A common goal for presenting data visually is simply to show what is important. A classic chart, made fun of in many cartoons, is of a businessman presenting a chart of sales, which typically nose-dives out of the frame in cartoon-world. The reason that the image is a classic is that it immediately shows what is important: Sales are going down – what else do you need to know? Action must be taken. Presumably the next chart in the presentation would show why sales are going down, but the basic point is that often the goal is simply to show the important information.

In a reflective/exploratory context, we may not immediately know what is important. It may be that we have many variables and are unsure which are important, or we want to know what is changing over time, or what variables are important for understanding other variables. In this case our goal will be more complex – we want to design displays that will allow us to discover what variables are important, whether by themselves or because of the effect they have on other variables.

Some characteristics of visualizations that have this goal are:

Simple charts. The goal is to present the important information rapidly. Extra explanatory information should only be added if it in no way clouds the overall message.

Vertical dimensions showing variables of high importance. The y dimension is generally viewed as a response, or dependent, variable. If a variable is important, we usually want to be able to affect it, so it should be put on the y dimension and explained by other variables. This leads to action – if we change x, then y will change.

Top-down story. If something is important, it is likely to be important across a large proportion of the data. Thus a design that starts by showing all or most of the data and drills down into different aspects of that overall visualization makes sense for most situations.

3.2.2 Seeing General Patterns

Much of statistics and data mining practice is concerned with understanding the main effects; typical models built for data will try to maximize the fit of the model – setting the parameters of the model so that as much of the data as possible is close

to what the model says they should be. In a time series prediction, the model tries to predict the value of one variable given the others. The fit describes how close the prediction is to the observed data. In a clustering model, a good fit will occur when data items are closer to other items in their clusters than they are to items not in their clusters.

Even when a model is not explicitly defined, we have a conceptual "sketch of a model" in our heads. When we do as simple a task as drawing a bar chart of counts of a variable, we have a model in our heads – the categories of the variable are distinct, and the counts have meaning. We expect that we will find that the counts have some distribution of interest; we would not, for example, try to draw a bar chart of people's addresses unless we had some special knowledge that might lead us to believe that they would not all be different.

For time-based data, models are of particular interest. It is rare that we would not at least be somewhat interested in predicting the future, and to do so, a model is necessary. We expect that variables will vary smoothly over time – or at least that values closer together will be more similar than values further apart. We want to understand trends and patterns: Are values increasing or decreasing over time? At what rate? Are they getting more or less variable, or is the variability stable? Are there cyclical tendencies that could be explained by a seasonal effect – daily, weekly, yearly? Are the relationships between variables changing over time?

For many data sets, the pattern of time events itself is of interest. As an example, the study of earthquake occurrences is a hotly researched area, with many models formulated for the space–time process (Ogata [83] outlines and compares several fundamental models). Are occurrences evenly distributed over time? If not, how are they related? Is there a correlation with other variables? With spatial locations?

A common goal of visualization is to understand the general state of a system, answering questions such as those posed above and summarizing rich data sets with visualizations that allow viewers to gain a sense of what is going on.

Some characteristics of visualizations that have this goal are as follows:

Aggregate elements Since the main effects are of interest, displays that aggregate data are to be preferred – means, measures of spread, or boxplots rather than individual values. Interval and area elements that summarize many data values should be used rather than points that create a glyph for each row of data.

Many views If there are many variables, views should be available that show these variables – even if the views are uninteresting. It is better to show a view that says "nothing is of interest" rather than to show no view and leave the viewer wondering if there is an effect.

Gestalt views Views that are complex, but show many variables or many relationships, are important. Since understanding is a reflective goal, it is acceptable for a visualization to need time to study and understand. Networks that show which

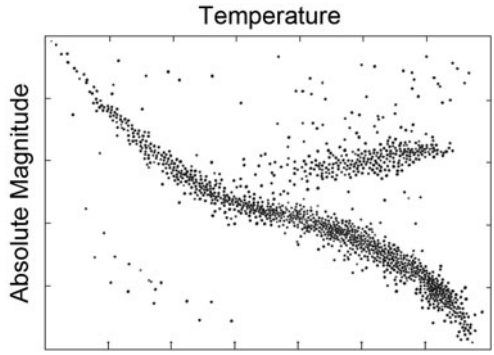


Fig. 3.1 Hertzsprung–Russell diagram. This famous scatterplot shows the relationship between temperatures of stars running *horizontally* and their absolute magnitudes (equivalently, their luminosities), shown in the *vertical* dimension. The long *diagonal structure* is termed the main sequence, the group at the *top right* are giants (with supergiants above them), and the group in the *bottom left* are white dwarves

variables affect each other ([139] – an example in this book is Fig. 6.18); high-dimensional displays such as parallel coordinate displays ([61, 62, 115, 128]) and projection tours ([15, 30, 42, 43, 108]) are all good candidates.

3.2.3 Spotting Unusual Features

When the primary goal is seeing general patterns, spotting unusual features is often a secondary goal – if we have a general idea of what goes on overall, then any items that are different from that overall pattern are of interest. For example, a famous diagram that has inspired much research is the Hertzsprung-Russell diagram, a version of which is shown as Fig. 3.1.

This figure performs the main goal of showing major effects admirably – the main sequence is clear, and the group of giants is also obvious. The white dwarf stars to the bottom left are not as obviously a main feature, and even if we decide that they are a main feature, we have a number of other stars not fitting into these groups toward the top. As theories were proposed to explain the main clusters in this chart (from about 1910 to 1930), the unusual values also needed explanation.

The Hertzsprung–Russell diagram is an example of a single chart that shows both main effects and unusual values. This is a rare case. More commonly, visualizations will serve one or the other purpose, and a good technique is to pair visualizations up with one showing the main effect and another showing possible deviations from that main effect. As an example, in simple linear regression where we are predicting Y using a set of variable X_i, plots showing the prediction against each X_i show the main

effects, and residual plots showing the differences between the prediction and the actual values plotted against the X_i help reveal any anomalous structure that might be present.

It is common for time-based data to have unusual features that are important not just for analysts to use in modeling the main effect but that can directly lead to action. These include:

Parameter shifts A certain relationship may hold true throughout the whole data, but a shift in how that relationship is parameterized can occur. For example, if a company splits its stock, the relationships between stock price and other values may not change in type, but they will certainly have different values!

Change points Similar to parameter shifts, change points occur when a relationship changes in some fundamental way. This includes differences in type as well as degree.

Population changes Many models assume that the data being studied come from the same general population. But new types of customer may start shopping at a Web site, a new company may start selling in an existing market, or a new set of organisms may enter an ecosystem being studied.

Spotting changes is a particular concern for real-time or streaming data. Once a general pattern is known, it is often important to monitor the incoming information over time, so that changes of the above types can be detected. Security data for computer, network, and human activity form a clear example. In security, the main patterns need to be filtered out and the unusual features highlighted. Visualization systems need to be carefully designed so that they fulfill the goal of showing unusual subsets, diminishing the visual impact of the often overwhelmingly large main body of data.

3.3 Questions

In the GQM methodology, questions should be formulated so that answering them will achieve goals (or show that they are not being achieved). For example, if the goal is to increase software developers' productivity, questions such as "How much does each developer produce?" or "How is each project progressing compared to the schedule?" are reasonable questions.

For visualization, a graphic provides a tangible form of question –more precisely, the description of a chart forms the question, and adding data to the description to create a fully realized graphic answers the question. In the above example, the first question might be embodied as a bar chart showing features added by developer and the second by a line chart showing percentage completion by time, broken out into separate lines using a color aesthetic on the project. The number of questions that can be posed are limitless, so this section concentrates on broad classes of questions, giving suggestions on how to design charts to answer questions that fit the goals of Sect. 3.2 when the focus is one, two, or many variables.

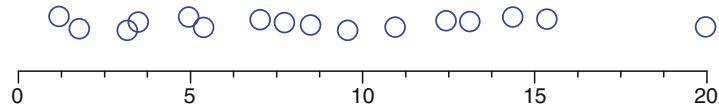


Fig. 3.2 Artificial outlier data. The data consist of 15 uniformly distributed data points and one outlier. The data are displayed as a 1-D dot plot, with vertical jittering to mitigate overlap

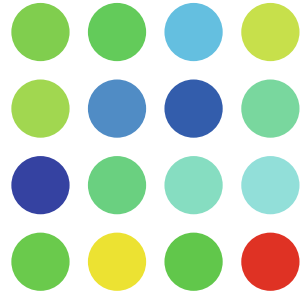


Fig. 3.3 Artificial outlier data. The data consist of 15 uniformly distributed data points and one outlier. The 16 data values are represented by *points* randomly arranged on a grid, and color hue is used to show the data values

3.3.1 One Variable: Unusual Values

The simplest types of questions that can be asked are about a single variable. Whether the variable is numeric, categorical, time-based, or of any other form, a common goal is to describe what the variable "looks like" in some general and useful sense. One common and important question that can be asked about a variable is: Are there any unusual values? In statistics, these are termed *univariate outliers*, and in any scenario, if you care about the variable, you care about unusual values of the variable. For example, if you are looking at a time series of measurements of network traffic, time periods with very low counts may indicate a problem with the hardware, and time periods with unusually high values may indicate a denial-of-service attack.

Figure 3.2 shows a simple display intended to answer this question. The values are shown as positions on the horizontal axis, and the unusually high value on the right stands out immediately. It is not a far outlier, but it is worth investigating.

Our visual system is highly effective at spotting patterns based on location. If all we needed to show was one variable at a time, there would be no reason to use any other mapping to display the variable of interest. Often, however, the variable that we would like to check is not the primary focus. The data might contain several variables, and we might want to focus on the relationship between A and B, say, but still monitor C. We may want to design a chart that uses position to show how A and B depend on each other, and find another way to show C. Figures 3.3 through 3.5 use aesthetics to show the variable we want to show unusual values of, with the positions of the elements randomly determined in this artificial example. It is a useful exercise to compare the effectiveness of the various mappings at detecting the outlier, a subject returned to in Sect. 3.4.

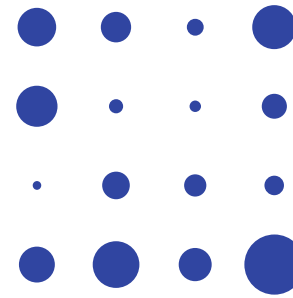


Fig. 3.4 Artificial outlier data. The data consist of 15 uniformly distributed data points and one outlier. The 16 data values are represented by *points* randomly arranged on a grid, and element size is used to show the data values

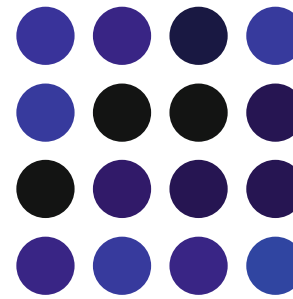


Fig. 3.5 Artificial outlier data. The data consist of 15 uniformly distributed data points and one outlier. The 16 data values are represented by *points* randomly arranged on a grid, and saturation is used to show the data values

3.3.2 One Variable: Showing Distribution

A second important question to ask about a single variable is: What is the distribution of the data? The previous section indicated how to answer a question about *unusual* values; this section asks about the *usual* values. Often there will be a general idea of what the distribution should be, and if that is the case, a good idea is to plot the expected distribution on top of the actual distribution. A formal statistical test, such as the *Kolmogorov–Smirnov* test, compares the observed distribution with a known distribution using parameters estimated from the data. Visually we can do the same, as shown on the right-hand side of Fig. 3.6. The three distributions' parameters are chosen to fit from the mean and standard distributions for the same data and are drawn on top of the histogram. It is easy visually to check whether the observed distribution fits one of these candidates.

Note that in Fig. 3.6a the distribution is so skewed that no details can be seen. This can also be the case when an outlier exists in the data. Often a figure like the skewed histogram will be observed when missing values are encoded using a number like 999,999 – a common practice. This is why checking for unusual values is presented here as the first step, and checking for distribution a second step.

Although statisticians in general are enamored with the *normal* distribution, in the realm of time series other distributions are common. *Exponential* distributions often model times between events well; the *Poisson* distribution fits counts of events occurring in given time intervals; if you check for arrival times given that a fixed number of arrivals have occurred, a good default assumption is that the times are

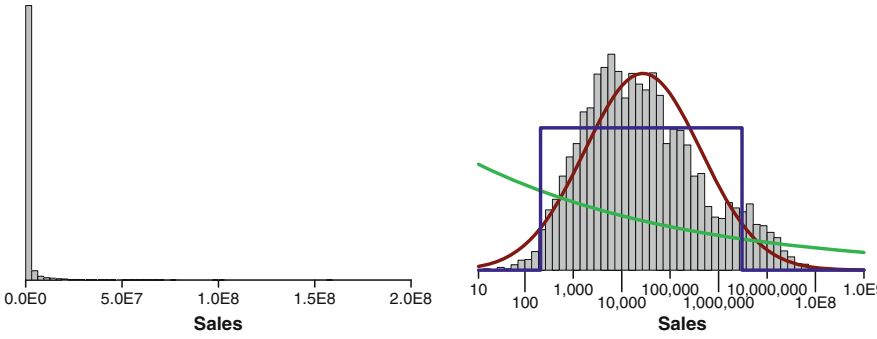


Fig. 3.6 *Left*: Sales of movie tickets as a simple histogram. The data are heavily skewed with most movies having small numbers of sales and a few blockbusters. To remedy this, the *x* dimension has been transformed using a log transform, giving the histogram on the *right*. Superimposed on this histogram are three distributions: uniform (*blue*), exponential (*green*), and normal (*red*). For each of these distributions the parameters have been estimated from the data

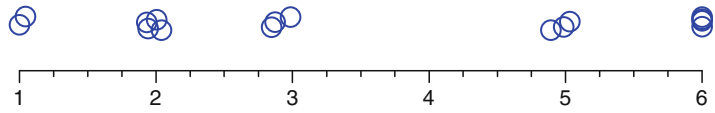


Fig. 3.7 Artificial two-mode data. The data consist of two groups of values. The data are displayed as a 1-D dot plot, with jittering to mitigate overlap

uniformly distributed. In fact, many of the less well-known statistical distributions (e.g., *Weibull* and *negative binomial*) are most often encountered when looking at temporal data or sequences. It is a good idea when analyzing time-based data to compare observed data to a wider variety of standard distributions than when looking at nontemporal data.

One question on the distribution that is often crucial is: Are the data all from the same group? This question cannot be answered completely using a single variable or display, but showing the distribution can reveal if there are multiple modes in the data – if the data form one or more groups. A histogram or other density estimate will reveal a multimodal structure well for a single variable, but if we do not have the luxury of allocating two positional variables just to show the distribution on one variable, how do we answer this question?

Figure 3.7 shows some artificial data with two modes with a simple 1-D dot plot. The two modes are clear – or, equivalently, there is an obvious gap in the middle of the data with no data having a value of 4. Figures 3.8–3.10 use aesthetics to show the distribution, and although the two modes can be seen in the first display (if not as clearly), the latter two appear to be of little help in spotting this important feature.

Section 3.4 gives some reasoning for why this is the case – the executive summary is that our visual systems can compare positions well, but when faced with

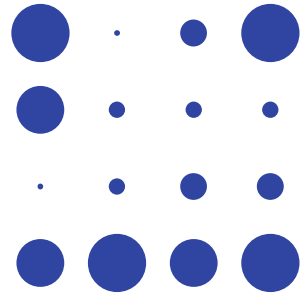


Fig. 3.8 Artificial two-mode data. The data consist of two groups of values. The 16 data values are represented by *points* randomly arranged on a grid, and element size is used to show the data values

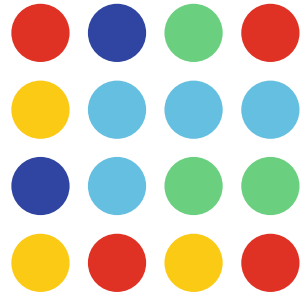


Fig. 3.9 Artificial two-mode data. The data consist of two groups of values. The 16 data values are represented by *points* randomly arranged on a grid, and color hue is used to show the data values

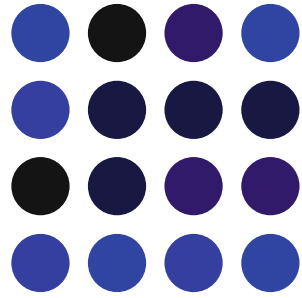


Fig. 3.10 Artificial two-mode data. The data consist of two groups of values. The 16 data values are represented by *points* randomly arranged on a grid, and saturation is used to show the data values

hues and other nonsize aesthetics, it prefers to make groups rather than compare values, and this grouping process is not consistent with the use we want – finding gaps or modes within a scalar range of values.

3.3.3 Two Variables: Showing Relationships and Unusual Values

Graphical methods are excellent tools for understanding relationships – there are a wide variety of possibilities that can be used to understand relationships even between only two variables. As shown in Sect. 3.3.2, spotting distribution features

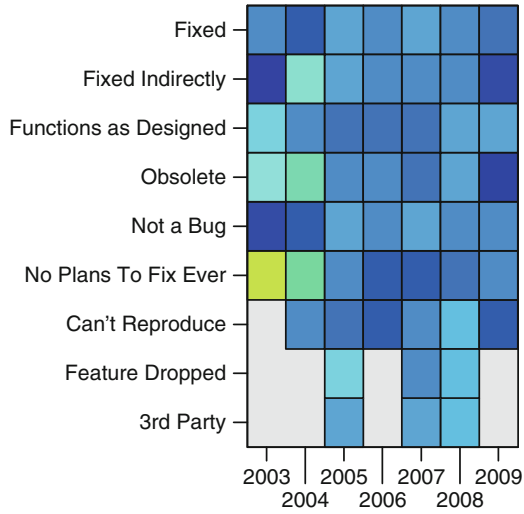


Fig. 3.11 Resolution of bugs by time and resolution type. These data are taken from an internal tracking database describing how bugs were resolved by year for a software development team at SPSS/IBM. A color aesthetic shows the difference between the actual count in the cell compared to what we would expect given the marginal distributions. *Cyan* represents "as expected," with *darker blue* representing cells with slightly fewer counts than expected. At the other end of the scale, *yellow* represents cells with more than expected counts. Empty cells are shown in *gray*

in one dimension is hard when using aesthetics – position is much better. When we are interested in answering questions about how one variable is related to another, the task is harder and so the conclusions apply more strongly. To answer questions about how one variable depends on another, by far the best approach is to map both variables to dimensions.

Given that approach, the decision on which element(s) to use to show the relationship is the next step. Chapter 2 gives advice on element usage. Specifically for showing relationships, the following form good starting points, depending on the data types:

Both variables are continuous: When both variables measure continuous values, scatterplots and line charts (especially when one variable is time) are excellent choices. Depending on the quantity being shown on the *y* dimension, areas might also be used. Note also that a continuous quantity can be treated as a categorical value by binning it (see Chap. 8 for details).

Both variables are categorical: In this case, the data can be thought of as a table, with each cell of the table potentially containing a number of items. Various tabular displays are possible (the term *heatmap* is often used for one such chart) with the most important decision being which statistic to display using an aesthetic for the element. If we displayed raw counts using a label aesthetic, we would simply have a table of frequencies. In Fig. 3.11 the cells are colored to

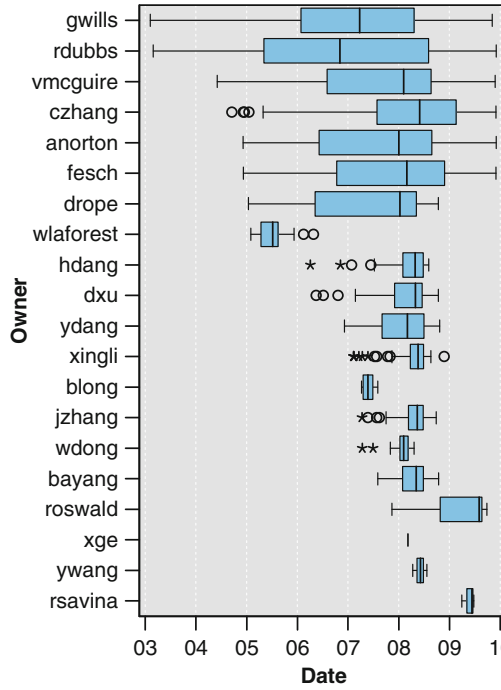


Fig. 3.12 Features and bugs. These data are taken from an internal tracking database at SPSS/IBM for the team responsible for working on visualization. Each row of data details a request for a feature or a bug needing fixing. Of the many fields available, this visualization shows the person assigned to the task on the *y* dimension (the task *owner*) and the date the task was created on the *x* dimension. A boxplot summarizes the dates for each person, and people have been sorted by the earliest date they worked on any task

show the difference between observed counts and expected counts. For this chart we are treating the continuous time variable as a categorical variable by binning values by year.

One categorical, one continuous: Here, a good solution is to show a set of univariate distributions side by side, one for each category. Dot plots can be used for small data sets, but for larger data sets some form of aggregation should be used. Figure 3.12 uses boxplots for a medium-sized data set (a few thousand values) and allows us to compare time distributions indicating who worked on tasks in the visualization team at SPSS. This display allows us to answer complex distributional questions. For example, we can see that, although for most people the distribution is skewed to the right (people do fewer bugs when they start working on the team), *wlaforest* is an outlier, taking on more tasks early on in his career.

People are very good at finding patterns in spatial arrangements. Ware [124] gives examples and explanations for the physiology of why that is, but the overall lesson is very simple: People see patterns rapidly when variables are mapped to positions. Figure 3.13 gives a demonstration of this. It contains four charts, each of which plots a pair of variables as a scatterplot. The data have been constructed so that each pair of variables in the plot has a correlation of 0.62. An automatic statistical procedure is very likely to consider the correlations between these variables to be

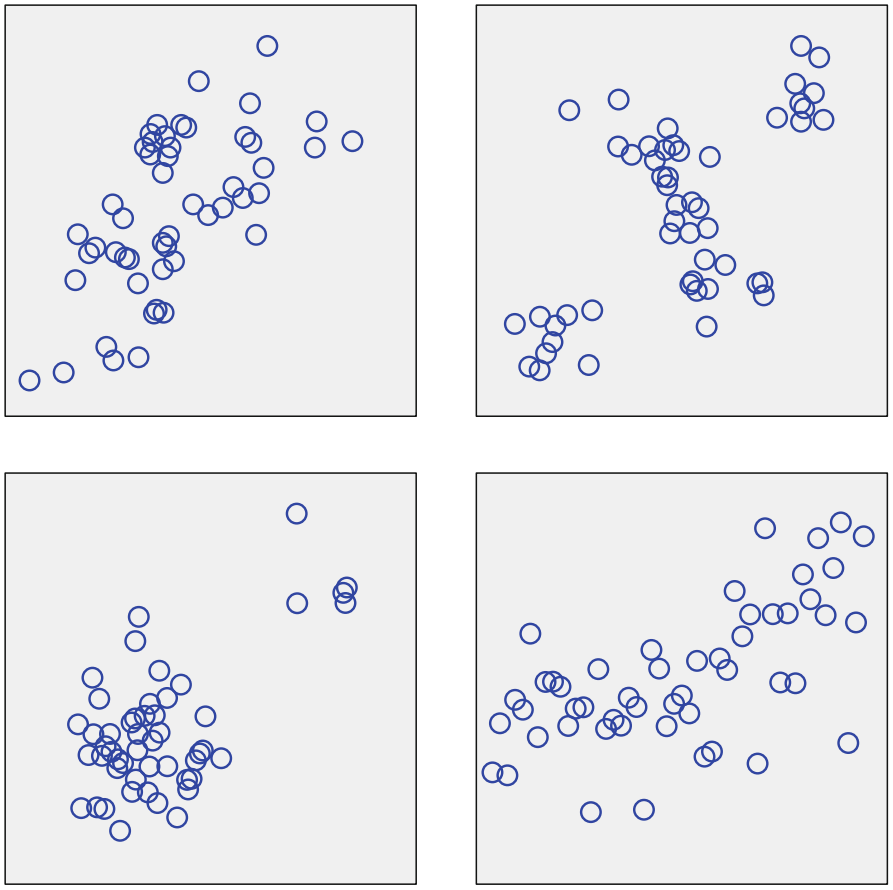


Fig. 3.13 Four pairs of variables with the same correlation. Each of these graphs contains a pair of variables that have the same correlation, 0.62. Each graph has 50 cases and was generated by a different set of rules. One set of data (*bottom left*) consists of a large group and a small group; within each group there is no correlation. Another graph (*bottom right*) is defined by the formula $y = x + x^2 + \varepsilon$, where ε is randomly distributed noise. Another (*top left*) is given by the formula $y = x + \varepsilon$, and the final graph (*top right*) consists of five equal-sized groups, with no correlation within the groups. This figure was motivated by a poster created for the Statistical Graphics Section of the American Statistical Association [2], which shows other "similar" correlations for very different relationships

the same, despite the very different structures they represent. The simple summary of a relationship by a numeric measure – any measure, not just correlation – should be investigated whenever time permits.[2]

[2]For a very *wide* data set, with many variables, it may not be possible to examine each pair of variables visually, and automatic techniques may be necessary. At that point, a suggestion would

The human capacity for pattern identification is not without issues, however. Many people, when looking at the top left panel of Fig. 3.13 will perceive groups, whereas the grouping arrangements shown appear only by chance. There is an important relationship between exploratory analysis and confirmatory analysis that this example highlights. It is a version of the standard scientific approach: Examine the information, derive a hypothesis, and then, with a new experiment, test the hypothesis. If we believe we see a relationship, and it is important to us, we should construct a test (on new data) to confirm our belief.

3.3.3.1 Trends: One Variable Varying Over Time

To display a variable varying over time, much of the advice of the previous section holds true. For a continuous variable, a time series line chart is a simple, common, and excellent choice. Time can also be converted into a categorical variable by binning, in which case the principles of the previous section can also be applied.

For categorical data and continuous time, it is not alway desirable to show time distributions as in Fig. 3.12, as that figure answers the question: How do time distributions depend on a categorical variable? More often, the categorical variable is dependent on time rather than the other way round. One solution is to use the categorical variable as if it were continuous, showing a scatterplot or even a line chart, although the latter should only be used if the advantages of emphasizing transitions between categories as time progresses outweigh the clumsiness of using a line (which implies a continuous change) for transitions between categories.

One situation that often occurs with time data is that data are given as two or more variables, but those variables represent aspects of the same quantity. Although we can use techniques such as those of Sect. 3.3.4 to deal with such data, it is often the case that the two variables can be combined into a single element, as in Fig. 3.14. The data used in this figure contain two variables for stock prices, one for the opening price and one for the closing price. We could show these as two superimposed time series, as in Sect. 3.3.4, but instead we can note that since we expect them to be closely related (the opening value for one month is very close to the closing price for the previous month), a better solution is possible. Here we combine the two prices into one directed arrow that goes from the start to the closing price.

---

be to try to calculate several different summaries so as to maximize the chances of detecting an interesting relationship. One such approach is the *Scagnostics* approach described in [136] and [137]. In this approach, nine different summary measures are defined for each scatterplot: *monotonic, outlying, skewed, sparse, clumpy, striated, convex, stringy, skinny*. These can be used in an automatic visualization system (e.g., *AutoVis* [139]) to present only the "interesting" relationships to a user, where "interesting" depends on the nine summaries, not just on a single measure.

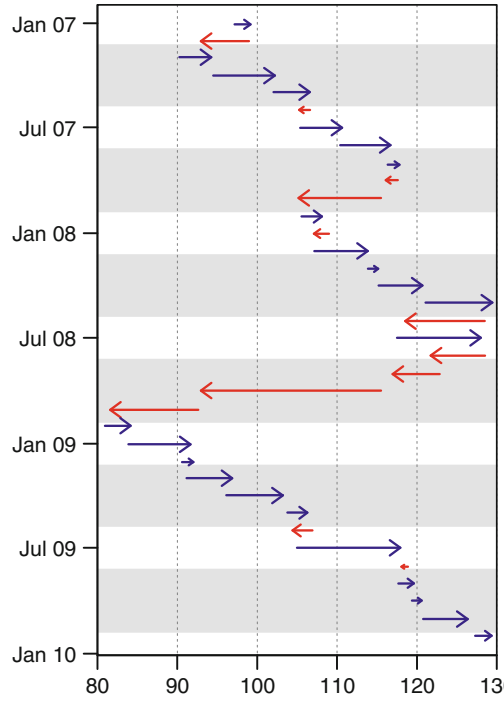


Fig. 3.14 International Business Machines stock price. The data for this chart are monthly open and close prices for IBM stock for the years 2007–2009. Each month is shown as an *arrow* (using the edge element) going from the opening price to the closing price. A color aesthetic has been added that encodes the direction the stock went in, with *blue* for increasing and *red* for decreasing prices. Note that a stock's price can change overnight – so the start of one *arrow* might not always be the same as the end of the previous one if there was a change in price overnight at month's end

3.3.4 *Multiple Variables: Conditional Relationships, Groups, and Unusual Relationships*

Figure 3.15 shows one of the simplest charts with multiple variables. Each line is a variable that represents the rank of a female name among all names registered by the US Social Security Agency [117]. An interactive, rich version of this chart can be found at [125] and is described in [127].

This chart encourages the user to group together similar names, where similarity is defined by having a similar popularity by year relationship. The biggest group is of names that were popular in the 1880s, declined through to the 1950s, and then had a comeback. Two names do not occur in this group. The first is "Elizabeth," which can be seen as a popular name throughout the time period. The second is "Madison,"[3] which was not a popular name for girls until quite recently. However, this is hard to see in the chart as displayed, as other names (e.g., "Abigail" and "Samantha") dropped off the popularity charts and then re-entered later. Because our visual systems see each line segment as a separate entity, we do not resolve

[3]Madison was one of the top 1000 *male* names in the first few decades of this chart, then died away in importance. When the female version of the name came into vogue, it had a small upswing again.

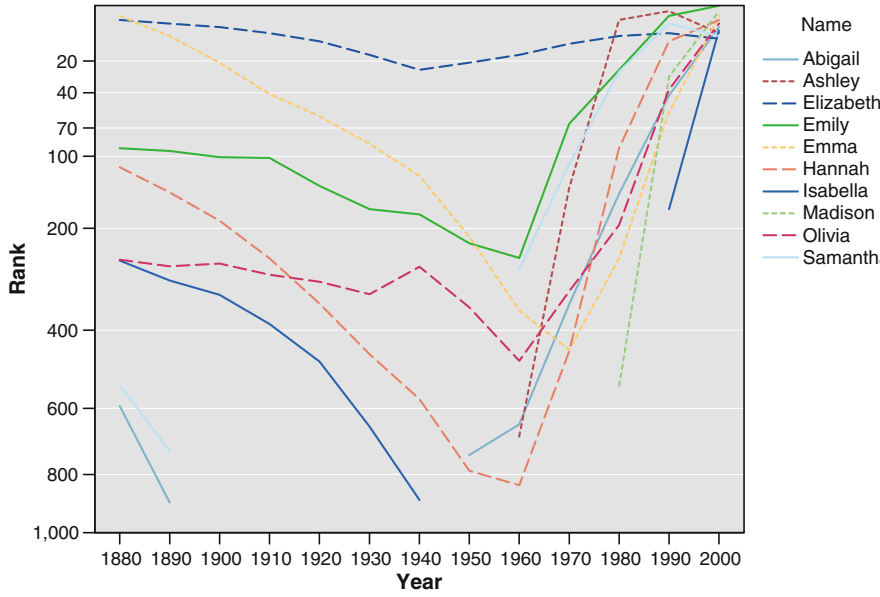


Fig. 3.15 Baby names by decade in the USA. The data for this figure were taken from the USA Social Security data site, where the rankings of children's names have been recorded by year and decade for over a century. This figure uses the decade-level rankings of girl names and has been filtered to show only the ten most popular names in the 2000s

the two segments for "Abigail" as a single item, and hence the difference between "Abigail" and "Madison" is not easy to see. A chart that faceted by name rather than using a line style would make it more obvious, at the expense of requiring more room to draw.

It is hard for people to think about relationships between multiple variables as one conceptual whole. What is easier and more natural is to think of *conditional relationships*. Such a relationship is one where we make a statement about how variable Y depends on variable X but state that the relationship is conditional on some aspect of one or more other variables. In Fig. 3.15 we make a grouping statement, saying, "For most names in the data set, popularity was high in the 1880s, dropped through 1950–1960 and then started rising again." We can then list the exceptions – the unusual names ("Elizabeth" and "Madison").

As shown in Sect. 3.3.3, position is the best way to show a relationship – especially when time is involved. Thus to show a conditional relationship we can:

1. **Add a third position dimension to the chart**: This is direct and simple, but our ability to judge 3D is very limited, so this technique is only suitable in a very few cases. A related technique is to use parallel coordinates to extend the number of additional dimensions possible; this can reveal structure for dozens of dimensions, but it needs an interactive environment to be most useful.

2. **Use a splitting aesthetic**: This is what we have done in Fig. 3.15 and works well for showing overall patterns, but when the number of groups created becomes large, it can be hard to see details. In other words, this is a good technique for seeing overall patterns, less good for seeing unusual relationships that may be hidden by the overall data.
3. **Facet the chart**: Creating small multiples of the chart, one for each category that you want to explore, is a powerful technique with the main drawback that it takes a lot of space. If that is not an issue, then faceting is the most powerful technique. Control over the levels of faceting and the order within each facet provides options for revealing information.

At this point, some readers may be wondering about showing a relationship conditional on a continuous variable – how can we use splitting aesthetics or faceting if a variable is continuous? The answer is to use some form of binning or aggregation to convert from continuous data to categorical data. Chapter 8 gives details on techniques for binning, but it is worth pointing out that time data are very amenable to binning since we are used to dealing with natural time bins such as days, weeks, months, and years.

A final point to make in this section is that interactivity is a good tool in this area. Given a chart that shows a relationship between two variables, we can make a selection on another chart and link the original chart by making an aesthetic that defines two groups – one for the selected data and one for the unselected data. A user can then try multiple selections (for example, picking in turn the most popular baby names from a bar chart of such names) and observe how they differ from the norm. This technique is most useful in discovering and describing outliers but has the strong advantage of working with a wide variety of base charts. The use of animation is essentially an automated form of linking using a visibility aesthetic. Each time slice is selected, and the results are shown on a chart that only makes visible the selected data; the rest is hidden. If you are designing a chart where animation is important, then a good idea is also to provide a representation of the variable over which you are animating – in effect, showing the linking as well as the base chart.

3.3.5 Multiple Variables: Showing Models

When the number of variables becomes large, looking at combinations of variables becomes increasingly hard. Apart from the fact that the number of combinations grows rapidly, a more serious issue in practice is that although a data set may have many variables, combinations of them may measure the same or similar concepts. In linear regression, this is know as *collinearity*, but the concept is broader than just an application to linear correlations.

For this reason, rather than trying simply to increase the number of variables in displays and understand the plots and trying to account for similarity between

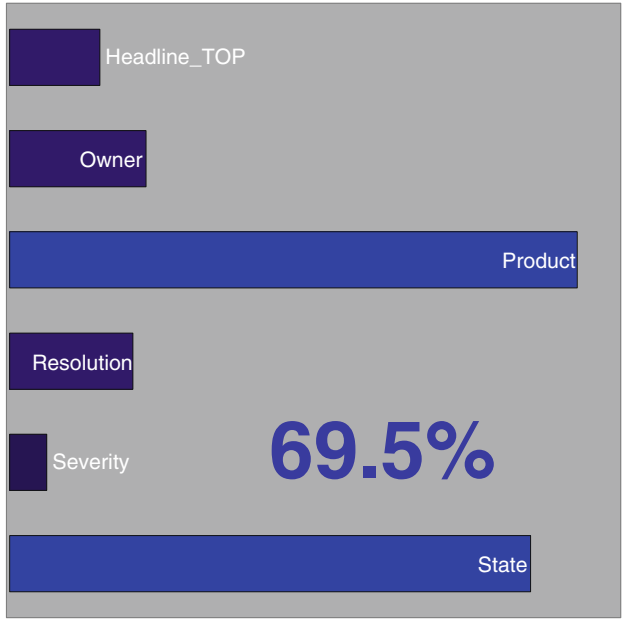


Fig. 3.16 Predicting the target version. This figure provides a simple summary of a random decision tree model that is predicting which version of the software was being targeted given data on the task being performed. The percentage value gives an overall measure of accuracy (0% would mean no better than guessing the largest class; 100% would mean perfect prediction). The bars give the importance of each variable to the model, using a permutation test to measure loss in accuracy when that column of data is scrambled

variables,an alternative is to build statistical or data mining models that will directly address the goal of the study. When that is done, the questions no longer concern the variables directly; instead, the questions are being asked of the model's outputs.

Consider the following example, in which the data consist of a set of tasks assigned to programmers on a single team. The goal is to understand the *target version* – the final shipped product: who works on the product, whether there are different types of product, and so on. As part of this study, a random decision tree forest was built to predict the target version. Random decision trees are described in [51] and were chosen since the data consisted of variety of measures (the date the task was created, free text task descriptions, measures of severity, categorical states) and random decision trees provide a robust way of dealing with such data.

Figure 3.16 shows an overview of the model. The variables that had any contribution to the prediction are shown as a set of bars, with the size of the bars showing the size of the contribution of each variable to the model; *Product* and *State* are therefore the most useful variables.

This figure is essentially just a number and a set of variables. The value of it is that the model produced is only one of many models that the automatic software

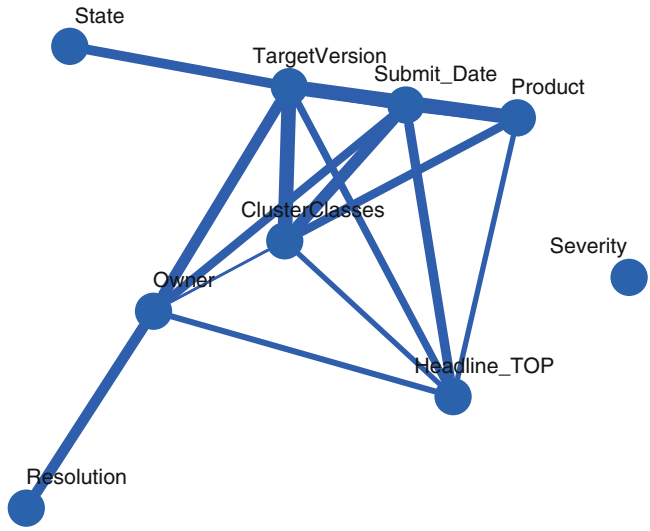


Fig. 3.17 Variable dependencies. This figure shows which variables depend on each other (an edge links nodes corresponding to those variables) and the strength of the associations (thicker is stronger). The layout uses a force-minimization technique that works a lot like multidimensional scaling

analysis system generates, and so the standard format allows a user to rapidly get the basics of the prediction – a rough measure of how well it performs and an idea of which variables are important to it.

This figure therefore provides an answer to the following questions:

- Is the model any good?
- What variables are used in the model?
- What are the most important variables?

These are fundamental questions for any model, and so this figure would work for many classes of supervised models, including cluster analyses, regression models, neural nets, or analysis of variance.

When this analysis was run, the analyst[4] was surprised to see that one variable was missing. This variable was *ClusterClasses*, which was created by a previous step in the analysis that clustered all the data using a two-step process involving k-means for a first step and then a hierarchical clustering step to produce final classes. In other similar analyses, that variable was included, so it was unusual not to see it in this model, prompting further investigation.

Figure 3.17 shows generalized correlations between variables in the model.Each correlation takes into account only a pair of variables, and the layout is similar in style to a multidimensional scaling layout. In this view, the target version is linked to the variables we saw in the analysis, but it is also linked to *ClusterClasses* and

[4]The author.

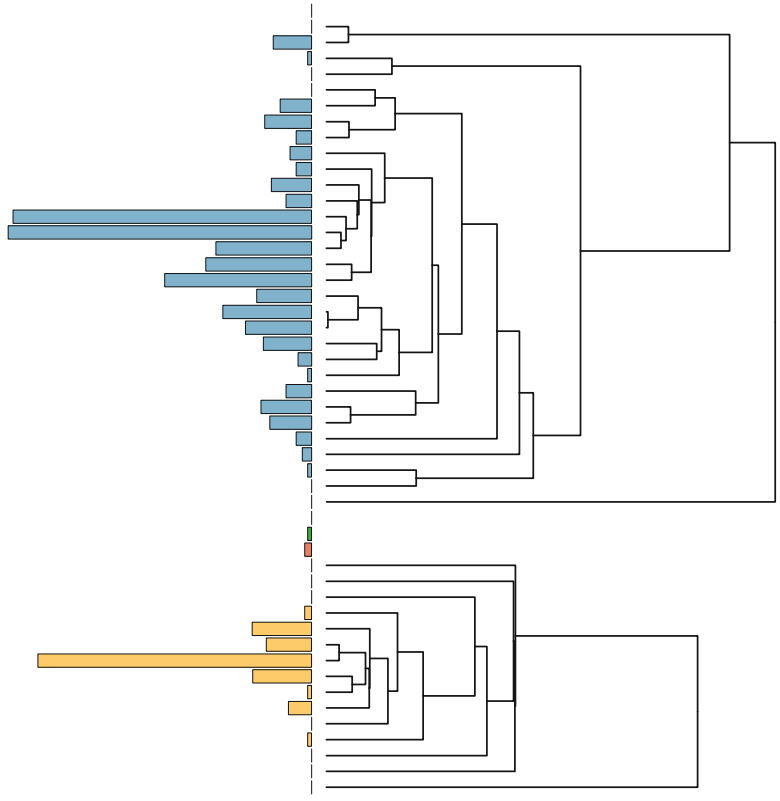


Fig. 3.18 Two-step clustering of all software tasks. The base data were clustered first into 50 groups using k-means. These groups are represented by bars on the *left* of the figure (bar size is proportional to the square root of the group sizes). These 50 groups were then clustered using hierarchical clustering resulting in two large groups and a few outlier groups (shown between the two major groups)

Submit_Date – the date the task was submitted. This figure answers the question: Which variables have some form of correlation?[5] – but Fig. 3.16 is based on a model that does not include a variable if it does not improve the prediction. Looking at both these figures, we next need to answer the question: What other predictor variables are related to *ClusterClasses* and *Submit_Date*?

Since *ClusterClasses* is itself the result of a model, the question resolves, somewhat recursively, to the same question with which we started this section, namely, understanding a model. Looking at a figure similar to Fig. 3.16 for the *ClusterClasses* model, we see that by far the most important determiner of the cluster is the *Product* variable. A direct view of the model (Fig. 3.18) shows that

[5]Since variables can be continuous, categorical, and even multiple-response, the "correlation" is actually a more general formulation as described in [139].

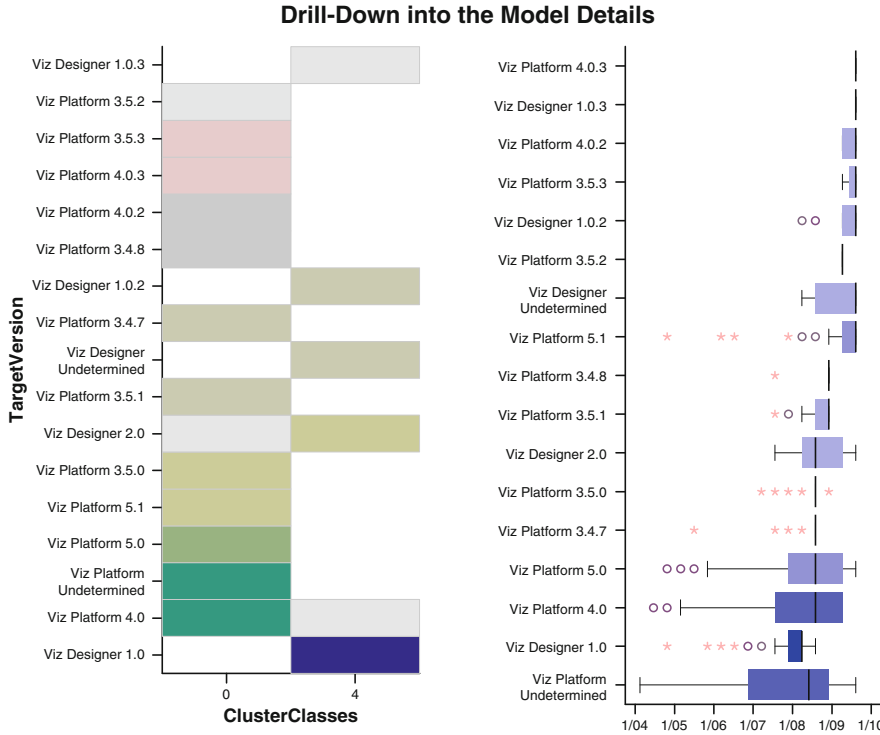


Fig. 3.19 These two figures show details found by drilling down into the edges shown in the variable-association view. *Left*: Relationship between *TargetVersion* and *ClusterClasses*. *Right*: Relationship between *TargetVersion* and the date the task was submitted to the system (*Submit_Date*)

the clustering resolves into two major classes, and further inspection shows us that those classes are highly correlated with *Product*.

To understand relationships between variables, we can apply the lessons of Sect. 3.3.3, which lead us to produce figures similar to those in Fig. 3.19.

These allow us to understand the model more clearly: The cluster classes are not useful for predicting the version since the clustering is essentially the same as the product. It is not as clear why *Submit_Date* is not useful – Fig. 3.19 seems to indicate it might be of value, but apparently some combination of the other variables makes it irrelevant. We could then return to the analysis process – modeling the submission date and seeing what affects it, and thus understand why it does not help us predict the target version.

Models and visualization go naturally hand in hand. Models can cope with many variables and make many decisions, but they need to be understood for action to be taken using them, and visualization is ideal for the task. On the other hand, visualization is a tremendous tool for understanding, but guidance is needed to choose *what* to visualize, and models provide that guidance. The basics of

previous sections apply – understanding general trends, spotting outliers, seeing relationships. Models provide new applications of our basic questions and help prioritize our questions.

3.4 Mappings

In Pinker's 1990 paper *A Theory of Graph Comprehension* [85], he states that a graph is a set of objects that tries to communicate a set of n-tuples of values on n mathematical scales, where visual dimensions correspond to the respective scales and the values on each dimension correlate with the values on the corresponding scales. In essence, the graph is defined by the mappings of values on scales to visual attributes. This approach, based substantially on Bertin ([8] reprinted as [9]) was developed in [25] with an emphasis on statistical graphics. In each of these cases, the authors define a graph in great part by the way it performs the mapping from data to representation.

Our visual systems have a lot of equipment – both in terms of detectors and in multiple levels of neural processing – that allow us to detect and make judgements based on positions. Sizes are also important, but beyond that there is less "hardware" support for aesthetics. Ware [124] is a good resource for understanding the physical basis for this, but rough-and-ready rules can be derived by remembering that the human visual system has evolved to achieve, essentially, a single goal – survival. Millions of years of evolution have tuned our systems so that the most critical tasks ("Notice something moving towards you"; "Classify this berry as poisonous"; "Distinguish a tiger from a deer"; "Work out the best route to run away from the tiger/toward the deer") are the ones that can be processed most rapidly.

Our visual system is therefore able to make good decisions on position (and knowledge of size helps a lot with position in three dimensions) and good decisions using multiple aesthetic qualities to group items into classes. Tasks such as locating the brightness of one object on a scale are not of strong evolutionary value; seeing that one thing is brighter than another and making groups of similar brightnesses is of much more value. So mappings that use nonsize aesthetics to portray values along a scale are not intuitive – we will naturally use our wired-in hardware to group things together.

Fighting our visual system is not a good plan; experimental evidence is clear that it determines our ability to spot patterns and answer the questions that we need to answer to achieve our goals. Time in particular has a number of properties that determine which mappings are most suitable:

- Time is continuous, but can often be used cyclically, and is naturally binnable at various scales to provide cyclically ordered categories.
- Time has a strong sense of direction.
- Time has strong emotional associations – as time progresses, things get older, they grow in size, they "yellow with age," among other things. These associations need to be respected in any mapping.

3.5 Systems of Visualizations

Section 3.3.5 gave an example of a system of visualizations – a set of graphs that together work toward a goal. It is a rare visualization that stands on its own; more often, a set of views are needed that show different aspects of the data and lead to a full understanding.

In Fig. 3.20 the goal is to discover which Beatles songs a listener likes best based on a set of available data (year, album, song length). This chart shows one aspect; answering a question on how much the ratings depend on the length of the song: Long songs are rated more highly, but the correlation is not strong.

Figure 3.21 shows ratings by album, with album dates shown using color.[6] A couple of albums stand out at either end of the distribution – *Yellow Submarine* is not liked, and *Magical Mystery Tour* not only is highly rated but also rated consistently.

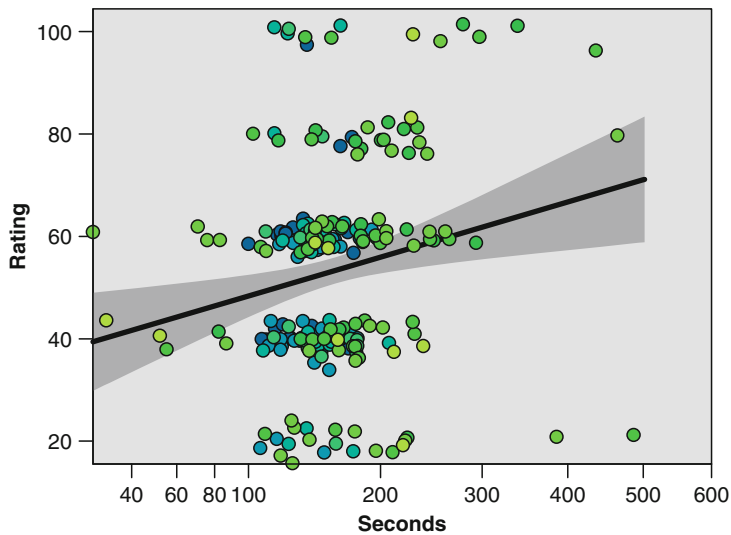


Fig. 3.20 Beatles songs, rating by length. The scatterplot shows a set of ratings of all the songs by the British pop band The Beatles. A linear regression *line* has been added to the data, with a 95% confidence interval that shows that the *line* has some significance, even if it doesn't explain much of the variability in the data. The color is used to encode the year of the song (*blue* is 1962, *yellow* is 1970)

[6]Note that the Beatles album *Past Masters* is a collection of songs spanning several years. Other albums, notably *Yellow Submarine*, also contain songs that bridge a year. In the figure, the mean year is chosen for the color, which works well for the albums that bridge a year but is less effective for the collection.

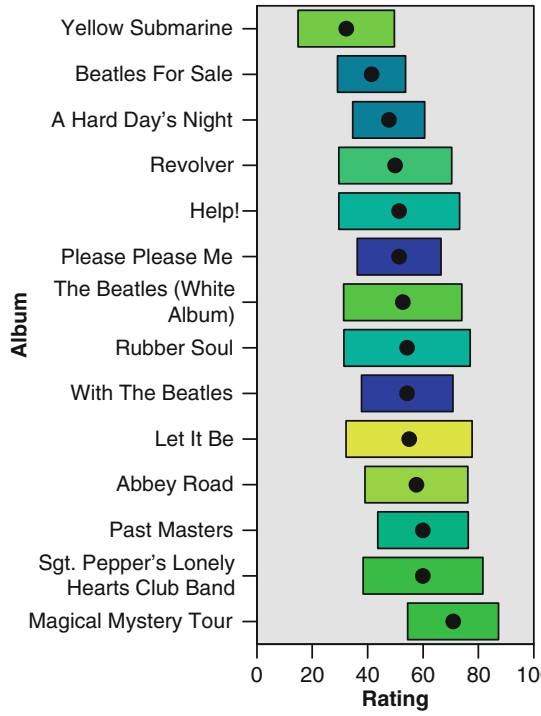


Fig. 3.21 Beatles songs by album. This figure shows song rating by album. The albums have been ordered by the mean rating for that album (also shown as the point element). The range *bar* indicates a 95% confidence interval for the mean, giving a measure of the spread of ratings for the album. The color is used to encode the mean year of the album (*blue* is 1962, *yellow* is 1970)

Figure 3.22 reworks Fig. 3.21, showing the mean rating by album and by year. This shows an interesting feature that follows from previous views – *Yellow Submarine* breaks up into two groups – the base songs from 1966 and 1967 that are rated positively, and a set of additional songs (added for the film) that fill out the 1969 album and are disliked. It appears that those songs are having a strong effect on the ratings for 1969 – as they are mainly instrumentals, perhaps they should be removed from the data set?

This example highlights a useful paradigm for designing information visualization systems: the concept of telling a story. The general idea is that as the user explores the data, she develops a narrative that describes the information she is learning, and that narrative forms a framework to describe results. Gershon and Ward [47] provide an early look at *storytelling visualization* and provide some guidelines on what makes for a good visualization narrative.

The technical details on how to build a narrative system involve ensuring that users can move smoothly from one view to another, that it is easy to modify existing views or move to related views. Some of the key requirements are listed below. Some concepts pertain only to interactive systems; others are equally applicable in a static representation of a narrative.

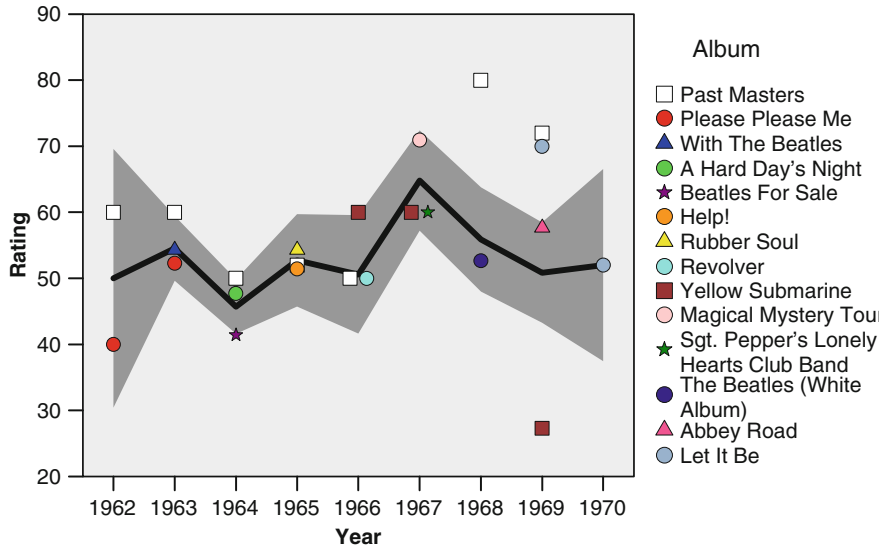


Fig. 3.22 Beatles songs by year and album. Each *point* represents a set of songs from an album that were originally released in a given year (the collection *Past Masters* is represented in six years, for example). A *line* gives the mean rating by year, with a 95% confidence interval for that mean surrounding it

3.5.1 Narrative Structure

Presenting visualizations as a narrative shares many of the same attributes as presenting a story using just words. Many books and essays have been written on this subject, and even summarizing them would take a considerable amount of space, so here are a few precepts that are directly usable.

- A good story should have a beginning, a middle, and an end. Similarly, a visualization story needs an attention-getting opening view, to give users a reason to delve into the middle, and should ensure that there is an end result in mind – the goal of the presentation should be clear in the final views.
- *Establishing the setting* is a critical task in a story. If a user is unsure of what the data represent, whether the variables are numeric or not, or any other basic details, they will not be able to understand charts presented to them that build on those details. Simple summary graphics help establish the setting for analysis.
- A major concern of story writing is *conflict resolution*. In graphical terms, this means that if charts present conflicting viewpoints (e.g., Figs. 3.17 and 3.18 on page 82), there should be a chart that resolves the ambiguity.

3.5.2 Consistency

When presenting multiple views, it is critical that the design be consistent. If color is being used for time, for example, then the mapping should be kept the same in all charts. As with all rules, this one can be broken, but if it is broken, then it is important to point out the change clearly.

Consistency is particularly important when using faceting, linking, animation, or filtering interactions. Each of these creates subsets of the data, presenting a different set of data in different charts. If the axes jump around, with the extents changing or the scale varying, it makes what is already a tricky task (spotting changes over time) much harder. A particular challenge is *streaming data* – data that are coming into the system over time. Because we cannot see what is coming in the future, we may start with a certain guess as to the range of data and then later find data that lie outside those bounds. Smooth transitions and minimal rescalings are needed to ensure that the resulting streaming visualization is usable.

3.5.3 Stereotypes

Judiciously applied, stereotypes have an important place in narrative. Our minds store information in this form, so that when we see things, we classify them by broad types. This reduces cognitive burden. Similarly, in a story, if we introduce a stereotypical character, we know that readers will understand them rapidly. In fiction, too much use of stereotype makes a story uninteresting – we read novels for *novelty*, after all – and the interest is in those characters that buck their stereotypes: Jean Valjean does not behave as inspector Javert expects a criminal to behave; Hamlet does not act like we expect a hero to; the main characters in *To Kill a Mockingbird* are defined by the ways they defy convention; no one in *Catch-22* acts like we expect soldiers to.

In visualization, when we have at least a conceptual model for what is happening, we can show expected behaviors rapidly; simple charts show expected distributions and relationships ("weight is normally distributed and strongly correlated with height"), but the focus should be on where stereotypes are defied. Unusual values or unexpected relationships make us re-evaluate theories, take action, or, at the very least, look for an explanation. Charts that make unusual features stand out make the visual narrative relevant.

3.6 Top-Down Versus Bottom-Up

In Sect. 1.3.1, we referred to a well-known "visualization mantra": *Overview first, zoom and filter, then details on demand*. This design technique is a *top-down* approach: The system presents all the data at a high level of aggregation

(equivalently, at a low level of detail) and allows the user to create subsets and refine what they are observing to gain more information. At a certain level of detail the goal is achieved, and so the quotation above, known as the *visual information-seeking mantra*, is well named. The approach *seeks* the result starting at the top and working down.

Critical tools for the top-down approach are good ways to zoom and filter using drill-down, animation, and filter interactions. Equally important is the third component – details on demand. In the top-down approach, users have little guidance about what to do. They start with the whole data at a high level and have to go searching in it. This will involve a fair amount of trial and error as the user tries a selection or zoom, discards it, and tries another. Rather than make the user complete that relatively intensive operation[7] regularly, some way of previewing what might happen allows us to keep our attention focused on the overview and not require as many false starts at zooming/filtering. Pop-ups that give basic information or give details that allow users to evaluate whether the item they are considering really is worth drilling into enable users to find the information they are seeking more rapidly. The mantra suggests that details should only be shown at the bottom level of the analysis, which seems overly restrictive. A better statement of the final part would be "always with details on demand."

Top-down has been the de facto standard design until recently. Data sets were relatively small, and techniques for aggregation were well understood for the standard tabular data that were the norm. Now, however, people often have goals that involve more data than can easily be presented in an overview, and data of a form that is not well suited to display as an overview. Further, the goal itself may be one that does not lend itself to a top-down approach, and a bottom-up approach is to be preferred.

An example of such data and such a goal is provided by search engines, where the data consist of every Web page on the planet, and the goal is to find a piece of information relevant to a certain search term. An interface that presents all the data on the Web and requires users to filter down to what they want is clearly inappropriate. What is required is a system that starts at the bottom – close to the information needed – and allows you to look around that area to find exactly what you are looking for. Rather than the top-down mantra, a more useful one would be:

Search first, widen and show context, then refocus.

[7] It may not seem that taking a second or two to drill down, see nothing of interest, and then return to the overview is much of a burden, but our visual memory is used to focusing attention ten times a second, and our short-term visual memory expires in about 300 ms. Anything that radically changes the view for longer than that disrupts our attention and forces us to re-evaluate the scene.

Kuhlthau [70] describes the information search process as taking six steps, listed below with a statement of the appropriate tasks for each state:

Initiation Recognize that information is needed and decide on the goal.
Selection Identify and select the topic of interest.
Exploration Investigate and find pertinent facts, locating features and orienting oneself.
Formulation Identify a focus, usually gradually through accumulation of concepts.
Collection Gather artifacts to define, extend, and support the focus.
Presentation Complete the process and prepare findings.

In the bottom-up approach, a system should allow a user to select a topic of interest rapidly and then allow exploration by expanding the search and formulating concepts. There should then be a mechanism that allows users to collect their results for dissemination. One interesting feature to note is that this process involves two methods of visualization – *exploratory graphics* and *presentation graphics* – that are often considered to be at odds with each other.

Kuhlthau's formulation is described in [54], which has two related chapters on visualizing search interfaces and visualizing text analysis. Text analysis often features a bottom-up approach. For example, the *phrase-net* technique [49] is one where a network of phrases is generated from a simple syntactical search (for example, "* at *") and the user can explore that network to understand relationships within a text document. Figure 3.23 shows a phrase-net view of this book, where we have searched for patterns of the form "* to *".

For time data, there is a very natural focus that can be used as an excellent default – the time period "now!". For many goals, we want to get some idea of the state of the system now or predict what might happen in the near future. So the present time is commonly of high importance. When presenting historical data, focusing on the present is a good default choice. The user should be able both to look at other time periods and broaden the window of time that is being looked at. Other focus-in-context techniques can be used such as adding a transformation to the time axis – for example, making a log transform on time, with "now" as the origin, or using a fish-eye transform (described in Chap. 9.2.3).

3.7 Summary

Designing visualizations is a process with several stages. First, the goal should be defined. Without a goal, visualization can correctly be characterized as "just a bunch of pretty pictures." Sometimes the goal is an explicit one, such as showing a result or describing a relationship. Sometimes the goal is to explore and learn something about a set of data. In either case, what is needed is criteria for determining whether the goal has been achieved ("Is the result clear?" or "Have we learned something new?") so that we can evaluate our success in design.

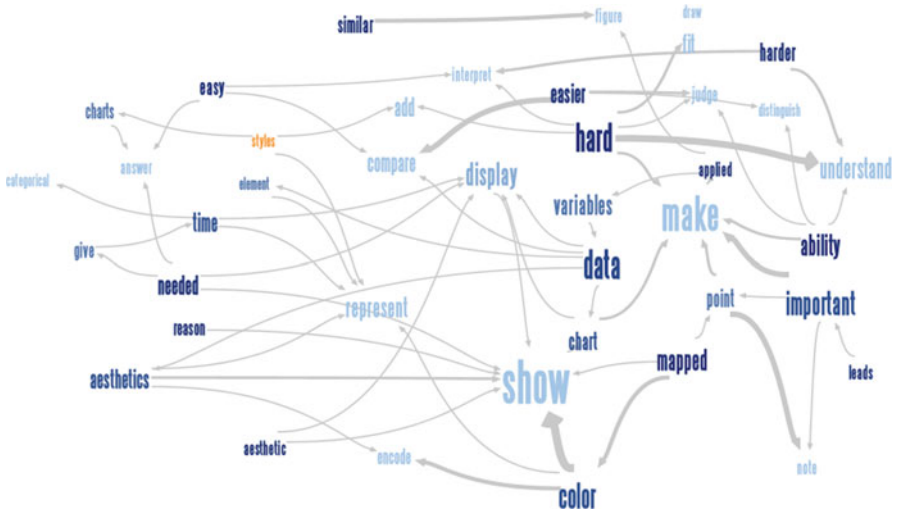


Fig. 3.23 This view shows the results of searching the text of this book for phrases of the form "*X* to *Y*." The resulting directed edges from the *X*s to the *Y*s are shown as a graph. Only the top matches are shown. The user can modify the matching criteria and expand/contract the number of matches. However, details that gave context for the edges and interactions that allowed you to refocus on edges or phrases of interest would be needed in a full system for analysis; PhraseNet simply provides an initial view. PhraseNet is part of *ManyEyes*, which is described in [121] and can be tried out at [120]. This view was exported directly from *ManyEyes*

When the goal has been set, we form a set of questions we want to answer to support our goals. These will vary from data set to data set, but there are classes of general questions that are common to many situations. Graphical representations of the data can then be created that will answer the questions. Those charts are created using mappings from data to visual characteristics. Those mappings are explored in Chaps. 5–7 and are at the heart of designing a good visualization.

The process of creating a visualization system involves deciding on sets of visualizations to show and designing a flow between them. A decision has to be made between top-down and bottom-up methods. Although the central exploratory steps of subsetting and broadening scope are common to both approaches, the most critical view in a system is the initial one. An initial view should hook the viewer in the same way a good story hooks the reader in the first paragraph. In particular, if a visualization system is showing the results of a data mining or statistical model, it is important that the first view provide not just an overview of the model but a description of how good the model is and what relationships it represents at a high level.

Interactivity, described in Chap. 9, should be used to facilitate the user's navigation between the views. Context should be kept as long as possible, so previews and pop-ups should be as informative as possible without overwhelming the basic chart. Knowledge of the human visual system is a strong aid in this process, helping to avoid mappings and interactions that seem reasonable but do not fit with the way our visual systems are wired.

Visualization is an *enabling* technology – it exists to help other disciplines. Whether it is statistical or data analysis, search engine output, text analysis, time series or weather forecasting, calendar applications or scheduling, visualization is not performed for its own sake. The best visualizations answer important questions, and the best visualization systems support important goals. If the choice is between a pretty but useless graphic and a boring but useful graphic, then the latter is the one that will get used. Having said that, strive not to force users into that choice. A beautiful and useful chart is the ideal.

3.8 Further Exploration

There has not been a definitive book on the GQM method, making further reading in that area more of a personal preference. *Foundations of Empirical Software Engineering: The Legacy of Victor R. Basili* [10] is a collection that includes the GQM methodology along with other ideas of Basili that are seminal in the field.

To delve into statistical and data mining models, Hastie, Tibshirani, and Friedman's book *The Elements of Statistical Learning* [51] provides not only an introduction but a good deal of depth. From the point of view of visualization, the book provides plenty of examples of model visualizations. The authors' focus is not visualization, and they reproduce other people's work for some models, but the graphs shown provide a good starting point for model visualization.

The *Journal of Computational and Graphical Statistics* published a paper by Cleveland [25] titled "A Model for Studying Display Methods of Statistical Graphics." The article has a number of discussants and the combination of paper and discussion gives much insight into the design of statistical graphics, not only highlighting principles but also indicating the relative importance different experts give to them.

Chapter 4
Types of Data

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to heaven, we were all going direct the other way...

— Charles Dickens, *A Tale of Two Cities (1859)*

4.1 Four-Minute Mile, Day of the Week, Bottom of the Ninth

People have measured time for many years. In Sect. 1.2 we saw how important it was to prehistoric cultures to have systems for denoting the passage of the seasons. A good case can be made that calendar time is the earliest form of data that humanity has analyzed in the abstract. In the age of sail, the accurate measurement of time was of critical importance for trade and military use since the most accurate method then available of estimating a ship's longitude[1] depended on the difference between "local time" and time at a fixed location. Sobel [102] states:

> To learn one's longitude at sea, one needs to know what time it is aboard ship and also the time at the home port – or any other fixed location ... Since the earth takes twenty-four hours to complete one full revolution of three hundred and sixty degrees ... each hour's time difference between the ship and the starting point marks a progress of fifteen degrees of longitude to the east or west. Every day at sea, when the navigator resets his ship's clock to local noon when the sun reaches the highest point in the sky, and then consults the home-port clock, every hour's discrepancy between them translates into another fifteen degrees of longitude.

[1] *Longitude* denotes location around the globe, on the east–west axis parallel to the equator, whereas *latitude* denotes the location on the north–south axis between the equator and the poles. Latitude is relatively easy to calculate with simple instruments, unlike longitude. In terms of the spherical coordinate system defined in Sect. 2.5.1, latitude is *theta* and longitude is *phi*.

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_4,

The accuracy of this system depended on the accurate measurement of time, and so measuring time became a matter of national security (see [102] for many interesting details). In our current world, measurement of time is as important and omnipresent as ever; we keep diaries and set our PDAs to remind us of events, we agonize over split seconds at the end of basketball and soccer games, and we hope our planes, trains, and buses leave on schedule, hence the desire to visualize time data faithfully. But we cannot just lump all data with a time component into one bucket; as the title of this section implies, there are different ways of measuring time.

4.1.1 Scales of Measurement

Stevens [105] proposed that data can be "typed" into a number of *m*easurement scales. Stevens' work has come under some criticism (from [119] for example) but generally for not being sufficiently complete, not for a basic flaw in his thinking. Stevens defined four measurement scales, each of which is more "powerful" than the next, in the sense that all operations that are defined on one scale are possible on subsequent scales. These scales are as follows, in order from weakest to strongest:

Nominal Nominal data are data that simply name things. Nominal values can be compared for equality, but that is all. They cannot be sorted, added, or subtracted. Neither can nominal values have their magnitudes compared to each other. Since time is intrinsically sorted, nominal time data are not common, but we do often record nominal data *at* given times; one example would be recording who is speaking during a meeting. People's names are nominal; there is no natural order for names.[2] We cannot say that "Li Xing" is twice as much of a name as "Sarah" or vice versa. The only use of the name is as a unique identifier. It carries no intrinsic meaning beyond that.

Ordinal Ordinal data are data that can be ordered. It allows comparison not just of equality, but also of position. We can put ordinal data into a definite order. However, we still cannot compare quantities. Even if we know that the steps in a recipe (heat saucepan, cook butter and flour, blend in milk, add grated cheese) have a definite order, we cannot say anything about the difference between the steps. The best we can do is ask how many intermediate steps there are (assuming we can calculate that) and use that as a surrogate. Ordinal time data are more common. We often are not interested in the exact durations between events, just in their sequence.

Interval Interval scales allow us to compare the distances, or intervals, between events. This is the natural scale for time data, as we often want to make statements

[2]The common alphabetical order is an *imposed* order on names. There is no reason except convenience and tradition that we might not order names by length, first vowels, number of syllables, or any other system.

about the distances between times. Any time we are interested in the differences between times (which is very often), we want to treat time as interval. A consequence of an interval having meaning is that subtraction has meaning for interval data; in general, if we have time data, we can subtract one date from another and are left with a meaningful quantity ("elapsed time").

Ratio Ratio scales are the peak of sophistication in Stevens' system. As well as allowing intervals between values to be calculated, they also allow ratios to be calculated. As a consequence of this ability, zero has intrinsic meaning in a ratio scale. Common physical quantities such as length, speed, weight, and so on are measured with *r*atio scales, but time is not generally amenable to ratio operations. What is the ratio of `July 1, 1988` to `February 23, 2008`? To use ratio operations on time we must impose a zero point on the time scale, converting time to *e*lapsed time, the time taken since a given zero point. Then we can speak of time ratios – we can say that the world record time (measured from the race start to end) for the marathon is about 760 times as long as that for the 100-meter dash.

Stevens' classification is not the last word in the description of time data. Even a simple data set consisting of counts by days of the week poses a problem. We want to say that days of the week are at least *o*rdinal, but strictly speaking they are not, since they are cyclical; Wednesday is both 2 days after Monday and 5 days before it. Yet it confounds common sense to say the days of the week have no natural order. So to represent the full range of scales for time, we should add another scale – perhaps a "cyclical ordinal scale" so as to represent this form of data.

For our desire to visualize data, this degree of complexity is not necessary. We are not as interested in measurement scale as a tool in and of itself but as a means to an end. Time is intrinsically *i*nterval in nature, but we often promote it to *ratio* by converting it to elapsed time, or downgrade it to *ordinal* by ignoring or failing to record the actual times and instead considering only a sequence of events. Rarely do we downgrade it all the way to *nominal*. Rather than use measurement scales as a tool to limit choices and ensure valid computation, we want simply to use them as a tool to suggest visualizations and guide our representations. How then does the scale of measurement for data affect the visualization of time?

4.1.2 Form Follows Function

The title of this section is actually a misquote from Sullivan [107],with the actual quotation being "form ever follows function," but it is perhaps appropriate that the shorter, less ornamented version of his statement became the rallying cry for architects in the early 1900s. The credo states that practical utility is the most important feature and that decoration and ornamentation are secondary considerations. Later architects and designers took a stronger view, stating that not only was ornamentation not a prime consideration, but that it should not exist.

This architectural principle, both in original and stronger form, has clear parallels with the design of information visualizations. As an example, Tufte's [111] stance on minimizing nondata ink follows the strong version "no form without function."

When *e*lements were discussed in Sect. 2.2, one point that was noted was that lines were not usually appropriate for categorical time. By "categorical" we mean noncontinuous; the points in time are at discrete time points, and it does not make sense to consider times between two sequential data points. In Stevens' formulation, *o*rdinal and *n*ominal variables are both categorical, and so the form of a chart should follow the form of the data. If time is discrete and the intervals between points in time have little meaning, then we should use an element that is discrete and shows the nature of the data. A line element with an ordinal *time* axis contradicts itself. The data cannot be considered to vary smoothly over time, as time is discrete. However, the *line* element interpolates between data points and suggests a continuous transition. Even worse would be to apply a smoother to the data, enhancing the deception.

That is not to say that just because your data are collected at discrete points in time, you cannot use a line element. The fundamental question is not one of collection but of the underlying process. If I collect data on the *Dow Jones Industrial Average* every 15 minutes, it is perfectly reasonable to show that as a line, and add a smoother if I desire, because the underlying process is much finer than the collection method. On the other hand, if I record all stock sales of a rarely traded company, the answer is not as clear. Because there are no trades except the ones recorded, it is not as defensible to interpolate between them. However, I might consider that the trades are simply a measure of the perceived value of the company, which might change more continuously, and so maybe a line would be appropriate. One solution to consider is to do both – show the sales as discrete items using points, and superimpose them on a line element to indicate that the value of the stock can be thought of as more continuously varying.

A similar argument holds for aesthetics as for the use of time as a coordinate. Aesthetics such as *pattern*, *texture*, and *dashing* do not have strong natural orderings and so are used most effectively with nominal data. *S*hape can also be used with nominal data, but we do have the possibility of using restricted shapes that allow us to use it for *o*rdinal data also. Sunflower plots [28] are an example of restricting shapes to make the aesthetic more natural to use with an ordinal data set. A sunflower plot uses a symbol to code the number of data points at a given location, with more petals meaning more data points, as in Fig. 4.1.

This approach could be adapted for a variable representing time. If time were an ordinal variable with a small number of values, we could carefully choose suitable symbols to show time values using the shape aesthetic on points.

Size is an aesthetic that we could use for interval time, but size has such a strong effect, and we naturally compare sizes as ratios, so it is not recommended. Similarly, transparency and color components such as saturation and brightness are perceived as ratio scales and so they too are unsuitable to display interval time (but are of course suitable for elapsed time). In fact, no common aesthetic is well suited to display an interval scale, a topic we will return to in Chap. 7.

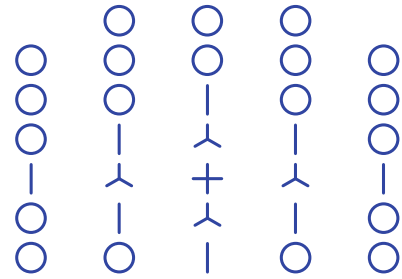


Fig. 4.1 Sunflower plot. The data are counts of numbers of observations at each location on a 7×5 grid. The *circles* represent a single item, *lines* represent two items, and the *three-* and *four-sided flowers* represent three and four items, respectively

4.2 Events and Intervals

Elapsed time has a very clear meaning. If I state that I ran a mile in 3 minutes, 5 seconds, there should be little confusion over the meaning of the statement (or, indeed, over the truthfulness or otherwise of the statement). For other types of time data the situation is not so obvious. If the data record that the temperature on a given day was a certain value, does that refer to a measurement taken at some point on that day, or is it meant to represent the whole day? More generally, when we discuss values at a given time, is that time a single point or an interval of time?

In this context, an *event* is an occurrence at a specific time that, at least in theory, has no duration. If our data are event data, then when we say "On Thursday it was 13° C," we mean that we measured the temperature at some unspecified time on Thursday, and at that instant, it was 13°. Often in this situation we do not know the exact moment of measurement. In Sect. 2.5 on page 49 we showed data on El Niño events, where the data gave various physical measurements on a daily basis. Without any other knowledge, we could assume that these data were event data, collected at some time on each day.

Because the range of days recorded is large, our lack of knowledge about when these data were collected during the day is not troublesome. A scatterplot (Fig. 4.2) makes it clear that if we knew exactly at what time of day the events were recorded, the plot would not change significantly.

After viewing Fig. 4.2 a natural next step might be to show it as a line, and add a smoother. Figure 4.3 is the result of doing so. Note two important differences:

- All but one of the gaps have vanished, and the remaining gap is much smaller. Our purpose was to interpolate and smooth the data at a daily level, but the smoother has also smoothed over gaps that represent missing data, making it appear as though the data were complete. This is a very common behavior for smoothers and can be a distinct problem.
- The smoother behaves poorly around 1987. In this region there are few data, and that causes the smoother to continue the upward trends around the missing data and infer high values for temperature at the edges of the gap. Only at the center of the gap does the smoother give up entirely and so leave a gap for missing data.

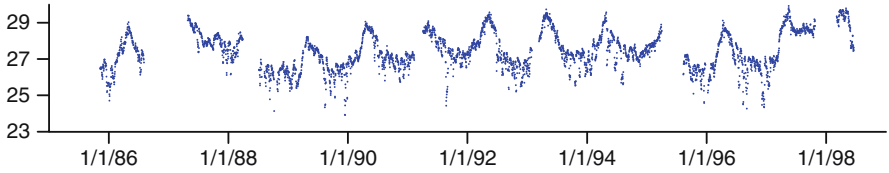


Fig. 4.2 Sea temperatures for the El Niño data set. The data have been restricted to show those that are recorded as having a longitude in the range $[-113, -107]$ and a latitude in the range $[4, 6]$. These likely correspond to readings from a single buoy

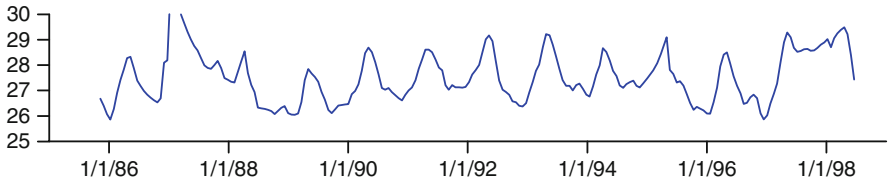


Fig. 4.3 Sea temperatures for the El Niño data set. The data are the same as for Fig. 4.2, with a loess smooth having been applied to the data and displayed using a *line* element rather than a *point*

Both these effects are due to the presence of missing data. The moral of these two plots is that missing data can make life difficult for even simple modeling techniques like smoothing, and when there are significant amounts of missing data, it is important to check plots of the original data and make sure that the missing data are not playing havoc with your algorithm.

If data apply to an interval of time rather than a single point in time, then we can treat them as event data if the intervals are small compared to the total time covered by the data. If the El Niño temperature data were in fact daily averages, there would be no difference in the way we might want to display them – at the scale of interest in the study (tens of years) there is not much difference between a point event and an interval of a day. If the ratio of the intervals to the total extent is larger, more care must be taken. Smoothers and other techniques that lead the viewer to believe that values vary within the intervals defined by the data should be avoided.

The choice of element for interval data is also trickier. Unsurprisingly, the interval element is a prime candidate. It explicitly shows a single value across an interval and so displays the results accurately. However, if we need to show multiple series, intervals are a poor choice as they tend to obscure each other. Lines and points are better in this case. For lines, if the software used permits it, the best solution is to use a stepped line, which keeps a constant value over the interval defined by the time and then changes instantly to the new value for the next data. Figure 2.3 on page 27 shows this variation in action. If this choice is not available, or if you want to show data as points, then the most common solution is to specify the center of the interval as the location at which to draw a point. Thus, if we want to draw a point for a given day, we pick "12:00 noon" as the time for that day at which to draw the point.

Often this is a trivial point. If you have a single series and you are only labeling the axis in days, then the choice of time to represent the interval of a day is not going to make much difference. It becomes more important if you have multiple time series at different scales that you want to display simultaneously. For example, we might want have a base series of event data consisting of trade prices for petrochemical companies and then superimpose other relevant series such as oil prices, international stock market indices, exchange rates, etc. on the base series for comparison. It is likely that the other series might be at a different scale and so we would need to make a choice (or at least understand our software's default) as to when to locate these series with respect to the more detailed granularity of the trade times.

4.3 Regular and Irregular Data

All time series analysts love regular data. Techniques such as first-differencing and ARIMA modeling (Chap. 8) expect data to be regular, and when we speak about interval time-data, the common assumption is that the data are regular and thus the intervals are all of the same length. *Regular* data are either event data, where the events are separated by a fixed amount of time, or interval data, where the intervals are all of the same duration. The great benefit of regular time data is that we can treat the time component as a sequence and reduce the time complexity down to a simple indexing:

$$i \rightarrow t_i$$
$$i \in \{0, 1, 2, 3, \ldots\}$$

In this case we end up with a simple series $\{t_0, t_1, t_2, \ldots\}$ for which a wide variety of classical techniques for analysis are appropriate. Moreover, the definition of missing data is much more clear; if we do not have a value at one of the regular steps, then that value is missing. This is in contrast to event data, in which we have no reason to expect data at any particular location, and therefore missing data must be explicitly noted as such in the data set.

Irregular time series, particularly when the data consist only of times, are often called *point processes*. In this book we do not use this term as we are overloading the term "point" enough already. Methods for the statistical analysis of point processes are typically very different from "standard" time series analysis, as will be seen in Chap. 8, but from a visualization point of view, they can be treated in a much more similar fashion, and so, unless noted otherwise, this book does not divide techniques into those appropriate for regular or non-regular series. Also worth noting is that time series literature defines a time series as *discrete* when observations are taken at specific, usually regular times.

4.4 Date and Time Formats

Taking a step back in the analysis chain, it can be hard even to get date information into a software system. If you are fortunate, you might have direct access to a database where the data have been stored as a *DATE* type (such as the various SQL data types: DATE, TIME, TIMESTAMP, ...). More likely, you are reading a value expressed as a string of text. Depending on the format, "January 8, 1919" might look like one of these formats:

`1919-01-08`	ISO 8601
`1/8/1919`	USA (in Europe this would be read as the first of August)
`8/1/1919`	European (in the USA this would be read as the first of August)
`8.1.1919`	German (many countries use different separators)
`190108`	Year, month, day as YYMMDD
`1919.008`	Year and day of year as YYYY.DDD
`19008`	Year and day of year as YYDDD
`5486`	Days since start of epoch (January 1, 1904 in this case)

Thepossibility of confusion between US and European style dates is well known. Both formats also suffer from the problem that sorting dates in this format as text values results in a wrongly ordered set of dates. To sort properly, dates formatted as text should be in *big-endian* form. Big-endian means that the highest-order-of-magnitude parts come first. When we write "154," we use big-endian format – hundreds first, then tens, then units. Neither US nor European formats are big-endian, and so are hard to sort. With one exception, the other dates in the table are an attempt to put the dates in big-endian format and so ensure that sorting by simple string comparison will put dates in the correct order.

Two important formats in the table are the first and the last entries.The first is an example of a date in ISO 8601 format. *ISO 8601* [63] defines a standard method for formatting dates and times. It has a number of variants, but the following formats are at the core of the standard for expressing dates, times, and date/time combinations:

Type	Format	Sample
Date	YYYY-MM-DD	`1919-01-08`
Time	hh:mm:ss	`14:22:06`
Date/time	YYYY-MM-DD**T**hh:mm:ss	`1919-01-08T14:22:06`

Note that the data hh hours are in 24-hour format, with 24:00:00 on one day being the exact same instant as 00:00:00 on the next day, but otherwise *hh* is in the range

[0,23], *mm* is in the range [0,59] and *ss* in the range [0,59].[3] There are variations that allow the separator characters "-", ":", and "T" to be dropped, and time zones can also be added, but as a rule if you are writing dates and times as text output that you expect to read in again later, then the above formats are rapidly gaining acceptance as well as having good sorting properties. As a bonus, because they are neither US nor European, no one can accuse you of any particular centrism!

The final format in the table is a little different. Technically it is not a date format, but rather a date *representation*. Many computer systems represent a date internally as a number representing an elapsed time from some arbitrary start point (often termed the *epoch*). Excel uses January 1, 1904 as its epoch and measures the elapsed time in days. Other systems do similar things. While this has clear advantages in simplicity, it can cause quite a bit of confusion. It is not a good interchange format as it depends on knowing two pieces of information (the epoch and the elapsed time units) that are not part of the data. There is also the well-known "epoch failure" problem – if we want to represent a date in the distant future or distant past, it might not be possible to define an integer large enough within a given computer language, causing potential failures. For example, the UNIX operating system uses the time 00:00:00 on January 1, 1970 as its epoch, and measures in seconds. Under this system, if we represent the time as a 32-bit integer, the latest time that can be represented is 03:14:07 UTC on Tuesday, January 19, 2038. Any time that is later than this will cause an integer value that is more than can be successfully stored in 32 bits. Typical behavior is that the number "wraps around" to become a large negative number, and so will cause programs to fail since they will see these times not as being in 2038 but rather way back in 1901. We leave as an exercise to the reader the task of calculating what date UNIX would think the quotation heading this chapter was from.

For these reasons, use of epoch-based formats is not recommended as a format for storing dates. Unless you have serious needs for a compact representation, use the ISO standard and maximize the chances of your data being read correctly by someone else.

4.5 Summary

Data preparation is a significant part of the analytic process. Sattler and Schallehn [94] argue that it takes 50 to 70% of the entire analysis time. It is therefore important to make sure that the data are read in correctly and that the time component is well understood. The goal is to make truthful and informative visualizations, and

[3] You might find a time where the seconds are actually 60. Although most likely this is an error, it is *just* possible that it is legal. On rare occasions a leap second is added to the calendar and is recorded as an extra second in a given minute, so on rare occasions a minute may be 61 seconds long.

if formats are confused, it can lead not only to wasted time and effort, but to misleading or plain wrong charts. Simple tricks like using a big-endian format for time data stored as strings make sorting easier, and the use of standard formats maximizes the chances both that other people will be able to use your output data and that you can be sure what your data are when you come back to them and not worry about the various possibilities of "02-11-09". Understanding more subtle differences like those between point time events and intervals of time, and how elapsed time differs from points in time, provides guidance as to the best representations of data, which leads back to our goal of visualizing time as usefully as possible.

4.6 Further Exploration

Most introductory texts on time series (e.g., [17]) will have some material on the different forms of data, but it is hard to point to sources that help with specifics. The documentation on the ISO standard [63] is interesting but technical. The same is true of computer references for dealing with date-formatted data.

Stevens [105] provides a good foundation on scales of measurements, but there has been a lot of discussion, much of it heated, since his original work. Rather than review the numerous technical papers, a better suggestion is to look at Chap. 5 of [80], or the excellent critique/review of the subject as applied to statistical data presented in [119]. The latter cites many good references for the deeply interested.

Chapter 5
Time as a Coordinate

Time has no divisions to mark its passage, there is never a thunder-storm or blare of trumpets to announce the beginning of a new month or year. Even when a new century begins it is only we mortals who ring bells and fire off pistols.

—Thomas Mann, *The Magic Mountain (1924)*

5.1 Put It on the Horizontal Axis

When I started writing this book, I would often have a variant of the following conversation with friends:

Friend: *So, what has you so busy all of a sudden?*
Me: *I am writing a book.*
Friend: *What's it about?*
Me: *It's about how to make effective graphical presentations of time-base data.*
Friend: *(looking puzzled) Don't you just put time on the X axis?*

A lot of the time – perhaps even *most* of the time – this is indeed the basic technique you will want to build on. Put the time component on the x dimension, and then work out what to do with the rest of the data. There are fundamental reasons why this is so.

Cleveland [21] presented a list of ways to encode a continuous variable, ordering that list by how effective it was to make comparisons between values based on the encoding technique. The list is as follows:

1. Position along a common scale
2. Position along identical, nonaligned scales
3. Length

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_5,

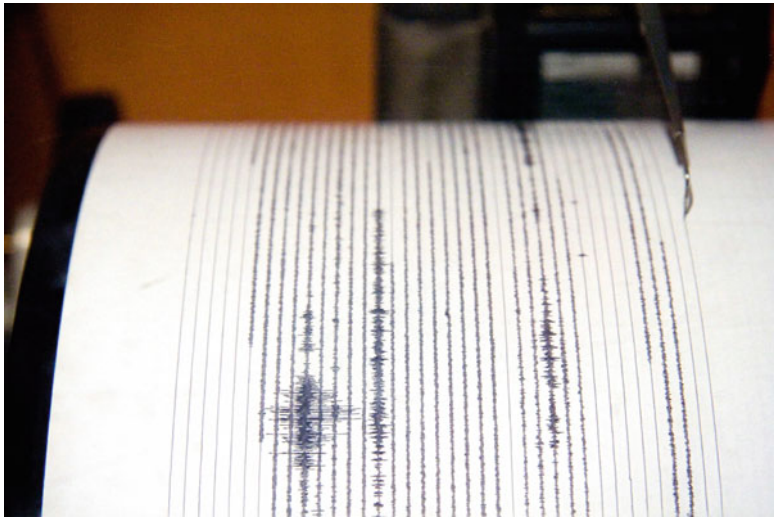


Fig. 5.1 Drum seismograph. The drum rotates constantly over time. The needle moves transversely from *left* to *right* with a fixed speed, generating a line that winds around the paper roll. Seismic activity makes the needle vibrate about the constant spiral line. Toward the beginning of the roll (*bottom, left of middle*) you can see a period of strong activity where the oscillations overlap earlier and later traces

4. Angle/slope
5. Area
6. Volume
7. Color

Position is the most useful of these mappings for encoding continuous data. Since time is fundamentally continuous, if it is one of the two most important variables you wish to display, then it makes sense to use it as one of the coordinates. Since we usually think of values as depending *on* time, rather than time depending on something else, time is almost invariably placed along the horizontal dimension. The standard time series chart shown in Fig. 2.2 on page xxx is standard for a good reason: It has strong theoretical properties.

Practically speaking, it is also an easy chart to construct. If we are observing data over time, they come to us in time order, so if we want to track a value over time, we can simply start at one end of a roll of paper and move along the paper as time goes by, marking observations as we see them. If we automate the process of moving the paper by winding the roll around a slowly rotating cylinder, then we trace out a time series plot simply by having a recording device move sideways on the cylinder as the value being measured changes. Seismographs (devices for measuring and recording movements of the ground, such as seismic waves associated with earthquakes) are an example of such a device, and Fig. 5.1 shows one from the mid-twentieth century.

Mechanisms such as seismographs,where changes in a measurement are written directly to paper, are one of the few examples where there is no intervening stage in the visualization process. Data are not even stored; there is a direct connection between measurement recording and visualization. Further, the seismograph was an important visualization and discovery tool. The traces produced directly motivated and stimulated theories of seismic activity and earthquake activity.

Sir James Alfred Ewing, Thomas Gray, and John Milne were three British scientists studying earthquakes in Japan. They founded the Seismological Society of Japan, and in 1880 Milne invented the horizontal pendulum seismograph [36]. This was the first device to measure transverse motions of the Earth as well as longitudinal ones. Robert Mallet[1] believed that earthquakes were caused by longitudinal waves only – waves oscillating in the same direction as the direction of propagation, out from the epicenter. In his detailed report on the earthquake of 1857 in Naples [74], he states:

> The shock, properly so called, that which shakes down buildings, &c., and the precedent and succedent tremors, are all waves of the same order, differing only in dimensions, and more or less in direction; the shock, being a wave of large amplitude, the tremors of small or very small; the latter also, in part, mixed with small transversals.

On March 8, 1881 the three scientists used their new seismograph to record transverse and longitudinal motions during a Japanese earthquake. Their device was not drum-based but instead a smoked-glass plate that rotated completely every 50 minutes so that the readings were written directly over each other. The pen rested on the plate near the edge of the disk. Thus, when there was no seismic activity, a circle would result as the disk rotated and the pen remained fixed. Seismic activity caused the pen to move radially – inward and outward from the center of the disk – causing the plate to display oscillations about that circle, as shown in Fig. 5.2.

The results disproved Mallet's assertion and started a whole new line of theoretic inquiry. From our point of view, Fig. 5.2 is important for two reasons. First, it demonstrates the power of a simple time series chart as well as the value of having a reference line at important levels. The basic time series chart is not only a presentation tool but has proven use as an exploratory tool, facilitating understanding and leading to discovery. The second reason for its importance is that it indicates a major use of time series – time-aligned comparison of data. In this example, both the transverse and longitudinal waves' magnitudes can be compared to each other and a clear correlation made visible. In Cleveland's terms, the time dimension is a *common scale*, and the two different measurements are made on *identical, nonaligned* scales.

[1]Mallet was a mathematics graduate of Trinity College Dublin, as was I. However, he graduated 150 years before I did and managed to do so when he was only 20 years old. He coined the words "seismology" and "epicenter" and was a major researcher in this area.

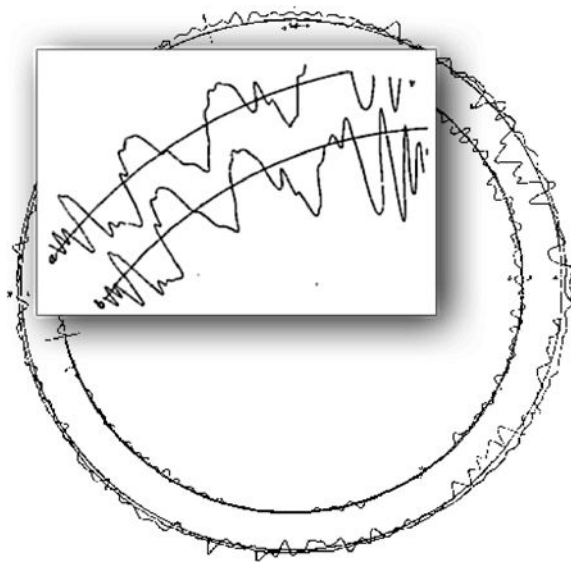


Fig. 5.2 Seismic activity as recorded by the Ewing seismograph. The main figure shows two circular traces indicating longitudinal and transverse waves as recorded on a smoked-glass plate. The *inset* shows details of a trace; clear correlation can be seen between the two wave components. McLachlan [77] provides the original source for this figure

5.2 Event Occurrences

In Chap. 4 the different forms of time data were discussed. One such form is *event data* – data that consist of events at (usually irregular) times. This can be considered as the simplest interesting form of time-based data – events for which the only data are *when* they occurred. Figure 5.3 shows an example of such data.

The data for Fig. 5.3 come from a study by Lewis F. Richardson, an expert in complex systems of equations and meteorology who became interested in the causes of war. He collected data on many wars and analyzed them, presenting his results in his book *Statistics of Deadly Quarrels* [91]. The main list of deadly quarrels was refined and tidied by Wilkinson [133], and the 315 cases identified by Wilkinson are the ones shown in 5.3. The data used in this section consist of the names of the wars, their magnitudes (number of deaths on a log 10 scale), the start and end dates, and the number of "pairs" – opposing groups within the conflict. A section of the data table is given below.

Case	Name	Magnitude	Start	End	Pairs
1	World War I	7.2	1914	1918	44
2	World War II	7.3	1939	1945	86
3	Taiping Rebellion	6.3	1851	1864	1
4	US Civil War	5.8	1861	1865	1
	...				
315	Bolivia	2.65	1952	1952	1

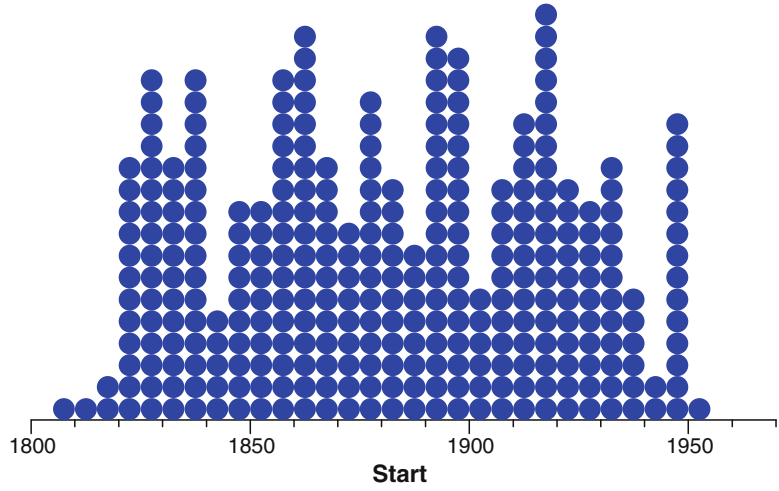


Fig. 5.3 Deadly quarrels by year. This stacked dot plot shows the starting year of each quarrel listed in Richardson's collection, with years binned by decade. A brief visual inspection shows no particular pattern, and Richardson's formal analysis concluded that deadly quarrels occur randomly over time with no particular trend or model

In the simple example of Fig. 5.3 we have ignored all data aspects except for the start time. This simple display allows the relative density of occurrences over time to be explored, but in addition, this chart serves as a basis for the construction of more informative plots. The display is intrinsically a 1-D plot – the stacking of points on top of each other is not an additional data dimension but is instead simply a visual device to allow us to see the relative density of the points more clearly.

In Fig. 5.4 we have augmented the basic view using *s*ize and *c*olor aesthetics to represent the `magnitude` of quarrels and their `duration` in years. Since we are not changing the *p*ositions of the element, we are leaving the focus clearly on the time aspect of the data; the aesthetics add to that focus, not replace it. Contrast Fig. 5.4 with Fig. 5.5. The latter figure changes the `magnitude` variable from being an aesthetic to being the *y* dimension. This completely changes the figure and makes it appropriate for a different goal; instead of showing the `start dates`, with the `magnitudes` and `durations` augmenting that view, it now has the goal of showing how `start dates` affect `magnitude`, with `duration` a possible secondary effect.[2]

Unfortunately for Richardson, the data did not lead to any insight concerning the reasons wars started when they did, although additional data he collected on social, economic, and other information on the participants did lead to some conclusions on nontemporal explanations for deadly quarrels. However, since the time component

[2]Another minor change: The filled circles have been changed to open circles to help mitigate overplotting.

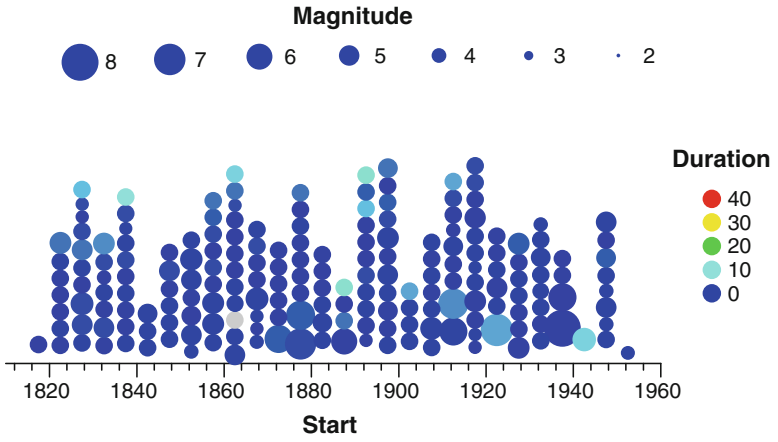


Fig. 5.4 Deadly quarrels: magnitude and duration, by year. This stacked dot plot shows the starting year of each quarrel listed in Richardson's collection, with years binned by decade. The magnitude of the quarrels, measured in deaths on a log scale with base ten, have been mapped to point sizes and the durations mapped to color. Even with these additional variables, the figure still shows no strong patterns that might explain why wars start

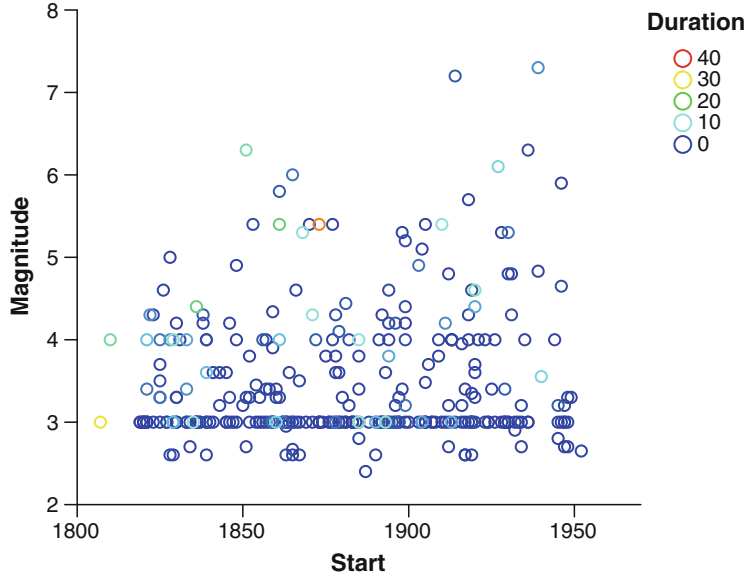


Fig. 5.5 Deadly quarrels by year. This scatterplot shows the relationship between the start dates and magnitudes of quarrels. The magnitudes are the most clearly displayed aspects of the data. The two world wars (with magnitudes greater than 7 – more than 10^7 deaths) can be seen at the *top right* of the figure. Another feature of the plot is the banding effect showing a bias toward recording magnitudes of exactly 10^3 and 10^4 deaths

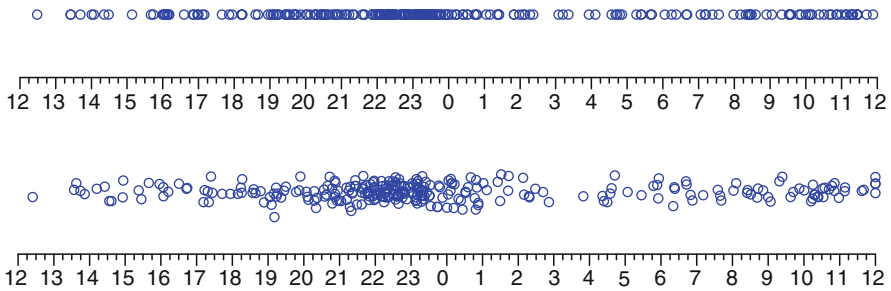


Fig. 5.6 Election night Tweet times. A sample of random comments made through the Twitter social networking site was collected during the night of the 2008 presidential election. In these views the times of the comments have been used for a 1-D point plot. In the lower chart, the points have been jittered. Many tweets were collected, and only those that contained an "interesting" key word, as explained in the body of the text, were retained. The times were recorded in the US Central time zone. In this time zone, John McCain made his concession speech at 2215, and Barack Obama made his speech around 2300

of Richardson's work was limited, we move to a somewhat more modern example – social network data consisting of short messages posted using the Twitter service.

The use of 1-D plots with a point representing each row in the data is the very first chart suggested in Robbins' book on creating graphs [92]; it is there called a strip plot, and Robbins presents a number of variations on it. Due to its fundamental nature, it is worth exploring in more detail.

Figure 5.6 shows a simple version and a jittered version of the 1-D point plot. Jittering helps mitigate the overplotting issue to a degree. As we noted before, this chart forms a good basis for building on. How might we continue, augmenting this chart to make it more useful?

Robbins suggests that the plot can be made more informative by adding a second dimension, making it a 2-D chart. The dimension would contain identifiers[3] for each case in the data, and so form a unique row for each data row. Of course, adding more data to a simple chart (if done at all sensibly) is bound to make it more informative, but in this case, a 2-D plot of identifier by continuous value is only of use for a small number of data points, as it allocates a whole line for each data item. For the 50 US states, it is a good plot. For our data set, in which we show a sample of around 300 "tweets" (short messages sent from users on the social networking site Twitter) over the day of the 2008 US presidential election, there are too many rows for this technique to be effective.

The second view in Fig. 5.6 shows a more scalable method also shown in Robbins: *jittering*. When we jitter, we simply add some random amounts to each

[3]The term *identifier* is used for a categorical variable that has a different value for each row in the data. It cannot be used for any analytic purpose, but it can be used to identify items in the data. An example would be people's names in a census or survey, or the text of Twitter comments in this example.

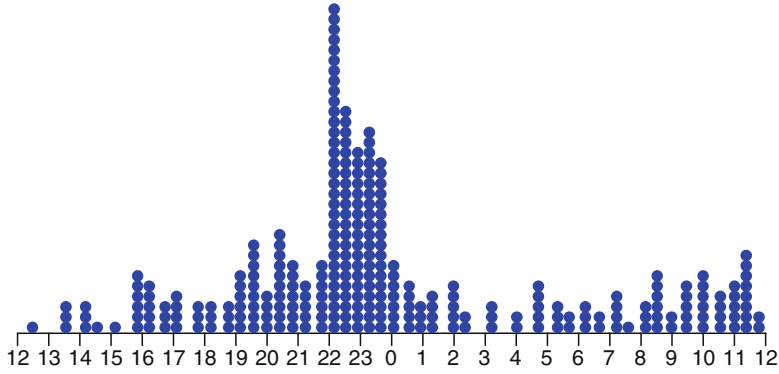


Fig. 5.7 Election night Tweet times. A sample of random comments made through the social networking site Twitter. The times have been binned using the dot binning algorithm described in Chap. 8 and all values in the same bin have been stacked on top of each other

data point's location to move the points around and reduce visual overlap. Jittering is one of a set of different *position modifiers* we can apply to a chart to adjust locations in a way that improves visual separation at the expense of making actual locations less meaningful. The most commonly used position modifiers are as follows:

Jitter Elements are moved by randomly small amounts. In some variations they might move in two dimensions; in others only in one dimension. Also the random amounts can be Gaussian or uniform. Practically speaking, this makes little difference to the result.

Dodge Elements are moved apart until they do not occlude one another. When dodging, it makes sense to move elements in a dimension that is either not used (if the chart is 1-D) or in a categorical dimension as this will cause the least likelihood of confusing the user into thinking the graphic element should be located at the position it appears to be in.

Stack Elements are piled on top of each other. Because this gives the visual impression of one element that is the sum of several others, it is very important that if the element's size is used to display a statistic, then that statistic must be summable. Stacking bars that represent counts, sums, or percentages are fine, but a stacked bar chart where bars show average values is generally meaningless.

When the simple and jittered charts fail to show enough information, using *dodge* or *stack* is the next simplest technique to try. For the Twitter data, we do this by first binning the data using the *dot binning* algorithm described in Chap. 8. This statistic's behavior is to place all points within a bin at the same 1-D location, so the stack position modifier is then needed to stack them vertically, resulting in Fig. 5.7.

This figure is an excellent starting point for further visual exploration. It shows over 315 data points, organized by time, clearly showing when events occurred – in this case a flurry of activity around the time of the two big election night speeches. Because it is simple, we have a lot of freedom to use other techniques to show

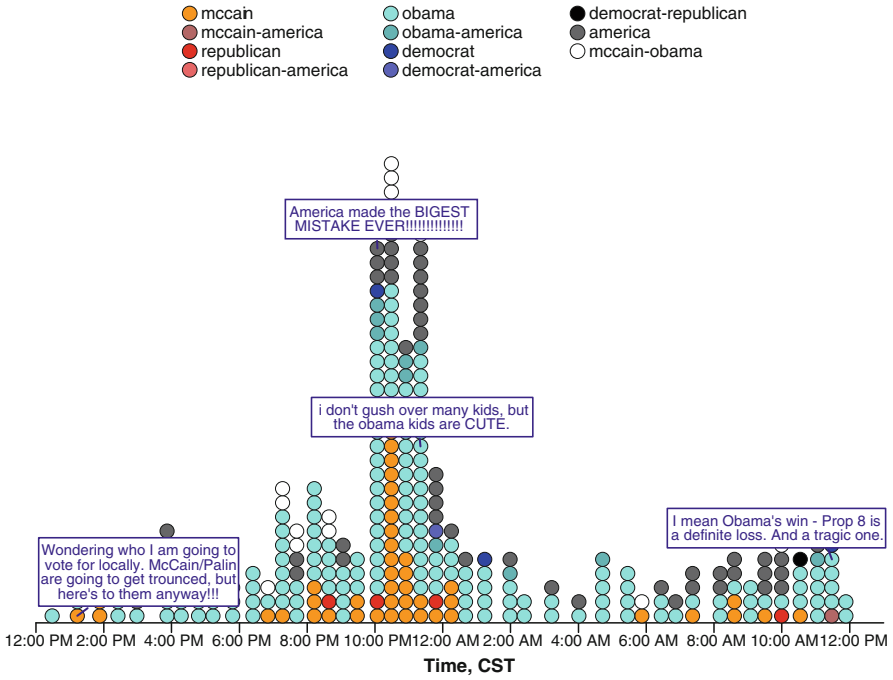


Fig. 5.8 Twitter election night topics. An enhancement of the base event data figure that adds a color mapping for message keywords and pop-up "tool tips" showing the full text of the Twitter messages

different data variables. This is an important point – when the base structure of a chart is good (*p*ositions and *e*lements), it makes adding other components like *a*esthetics, *f*aceting, and *i*nteraction much more likely to result in an understandable and actionable visualization.

The other variables that were collected for this data set include the text of the `message`, the `user` who sent the message, and some user-supplied information such as the user's `location`. Many fun and informative charts can be made based on the simple stacked dot plot. To cap this section on event visualizations, Fig. 5.8 provides an example of one of these. For this figure the goal was to understand what people were talking about, so the `message keywords` were displayed via a *c*olor aesthetic. These keywords were the ones used to reduce the many thousands of original sampled tweets to the 300 or so displayed here. The keywords were defined using the following procedure:

- All words in all messages were collected and listed in order of frequency.
- The most interesting high-frequency words were tagged: `america`, `democrat`, `mccain`, `obama`, `republican`.
- Only messages with one of those keywords present were retained in the data set, and the full combination of keywords was added as a variable for each tweet.

The derived message keyword variable was mapped to *c*olor and a color mapping chosen that made similar keyword sets visually similar. In the resulting figure we see mixed talk prior to the McCain concession speech, with the speech at 10:15 pm (2215 in 24-hour format) giving rise to many comments featuring `mccain`, followed shortly afterward by a burst of `obama` and `america` comments during Obama's speech. After the second speech, most of America went off to bed, leaving the rest of the world to talk almost exclusively about Obama until America woke up next morning (the posts from 12 a.m. until 6 a.m. are mostly from overseas).

A pop-up interaction (Sect. 9.2.2) was added so that moving the mouse over a point showed the message for that point. As it is not possible to demonstrate tooltips on a printed page, Fig. 5.8 simply shows a sample of such pop-ups drawn all at once. This figure is much more cluttered than the interactive version; it is hard to see the distribution of `keywords` under the multiple drawn labels. Pop-ups, as we will see in Chap. 9, are by far the preferred way of showing identifiers in charts with more than a few rows of data.

5.2.1 Many Events

For small numbers of events, individual symbols provide a flexible and powerful representation technique. However, suppose we want to look at a large number of events – for example, we want to explore a set of data on when people go to the movies in the USA. In 2008 well over a billion tickets were sold; plotting a symbol for each one would be tricky. Instead, we want to aggregate in some fashion. In fact, the data are reported already aggregated as weekly box-office totals. Even when aggregated to this level, we still arrive at a situation with nearly 7000 movie/week combinations.

We could try a number of ways of representing these data. One possibility would be a 2-D plot of `movie` by `time`, with dots at each intersection, sized or colored by the `attendance` variable. Unfortunately, we have too many unique movies for this to be a useful technique. Worse, since each movie lasts only a certain amount of time, the display would be mainly empty, with all the interesting data compressed into small areas.

We solve this problem by using the *stack* position modifier again. In the same way that the Twitter chart (Fig. 5.8) stacked individual items, we can stack aggregates to equally good effect. A simple chart that would work in most applications is the stacked histogram – each week shows a stack with the movies for that week stacked on top of each other. If that chart is not available, then you may have to bin the data outside the graphing facility and then pretend the bins are categorical and create a stacked bar chart.

When we have lots of different categories (movies in this example), the resulting chart looks somewhat disconnected – the stacked histogram is only a halfway step between the fully disaggregated state of having a graphic representation for each

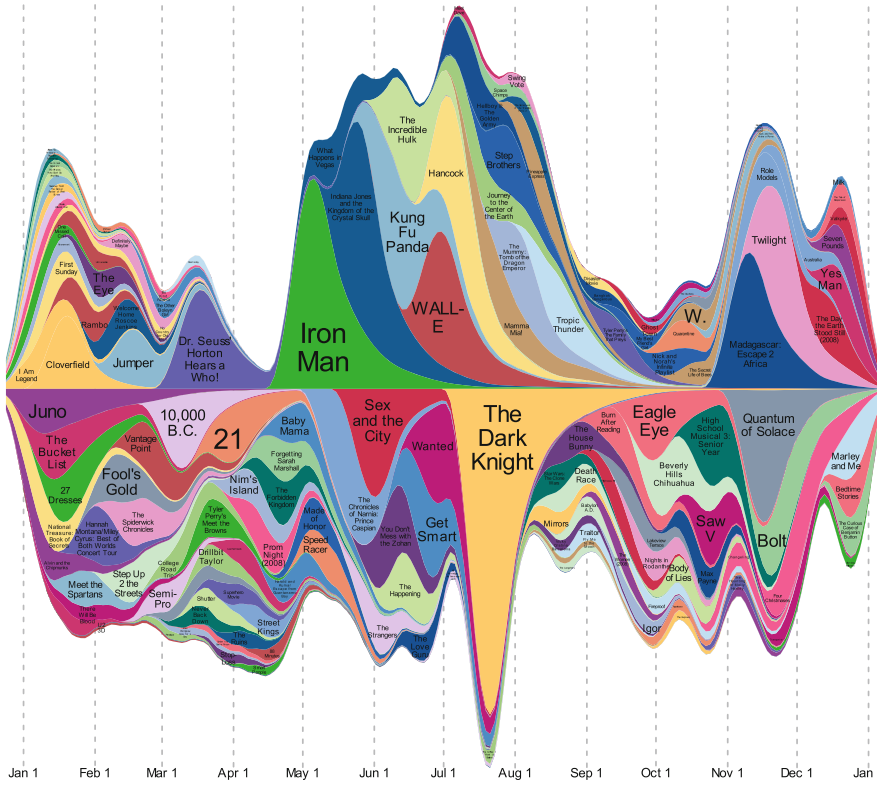


Fig. 5.9 US box-office receipts for movies in 2008. Time runs horizontally in this 1-D chart. Each movie is represented by a density estimate for when people saw the movie, and the densities are stacked on top of each other; the color used for each movie has no meaning; it is simply used to distinguish different movies. This figure is very compact due to space constraints; when stretched out four to eight times as wide, the riverlike nature of the chart is more apparent

ticket sold and the fully aggregated state of one graphic representation for a movie. The histogram breaks each movie up into a set of bars, and when we stack those bars, the fragmentation makes it very hard to see patterns on a per-movie basis – visually, the element dominates over the color, and so we see the data as a set of weeks, with each week split up into movies, rather than seeing the movies as the main item. For this reason in Fig. 5.9 we have replaced the histogram with a smoother way of estimating event density – a kernel density smoother. The smooth transition across time makes the time element less dominating, and we see each movie as a single entity.

This chart has an area element with a kernel density statistic, weighted by the box-office receipts (if we actually had data on every ticket sold, we would not need that feature, but we would probably need a bigger computer). The area element has a color aesthetic that shows the movie name. Since the movie name is a category,

this splits the area up, so we have one area for each movie. The color, since it maps to a movie name, has no real meaning – it simply allows us to distinguish areas easily.

The one extra piece that is needed in this chart is the modified stack technique. This is a novel position modifier that works in a similar way to the regular stack modifier, with the following changes:

- For each group (in this case, an area) the graphic shape is moved to the center and then stacked either above or below the existing shapes.
- For each group (`movie`) the algorithm chooses to stack above or below to make the total height of the chart as small as possible; this is basically a greedy "fit the next biggest thing in to grow the chart as little as possible" algorithm.

Just like a regular stack operation, the stacking occurs for each x value, so areas stack on top of areas naturally and total area is preserved, making area comparisons possible. This chart, despite its novel appearance, is actually a simple composition of existing techniques. It provides an example of the power of considering visualization as a language, instead of simply a set of types. For illustration, here is the central fragment of the VizML specification of this figure, fully describing the area element. When creating this figure, a number of different kernels can be used depending on the effect desired (a uniform kernel, for example, makes more of a steplike chart, which arguably reflects the underlying data better), and the proportion can be adjusted to control the degree of smoothing desired.

```
<area positionModifier="centerStacked">
    <densityStatistic kernel="triweight" proportion="0.05"/>
    <color variable="movie"/>
    <labeling variable="movie"/>
    <x variable="time"/>
</area>
```

This technique was initially explored in [52] under the name "ThemeRiver" and has been applied to text corpuses, listening history, and other fields. This technique works well when the data consist of events that are bunched together in time for each category, and when there is a moderately large number of categories (20 to 2000 seems to work well, depending on the application). It is also one of the few coined names for a visualization that is descriptive; since this chart is useful when you want to visualize a time evolving set of themes, the name works well.

5.3 Regular Categorical Sequences

Pure time events are a strong competitor for the simplest possible time data. Another strong competitor is typed sequence data. Pure sequence data would just be a sequence of identical recordings – a boring data set that says that at every fixed interval, something happened. All these data sets are mathematically equivalent to the set $\{1,2,3,4, \ldots, N\}$ for some N and are of as much interest as listening

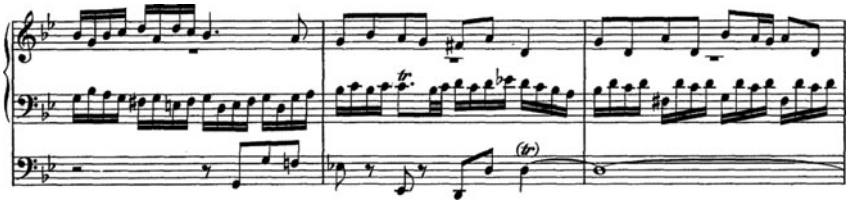


Fig. 5.10 A section from Bach's Little Fugue in G Minor, written between 1703 and 1707. The piece has four voices, three of which appear in this section, and was written for the organ

to a metronome. However, all it takes is to record one categorical variable at each point in the sequence to make it worth many millions of man-hours of study.

Figure 5.10 shows sheet music – a visual representation of a sequence of events. The simplest forms of sheet music show just notes in sequence, but to be useful, more is needed. This sheet music shows several constructs that we can immediately recognize:

1. Point elements represent notes in a time by pitch coordinate system.
2. The coordinate system is faceted into bars for different voices.
3. Glyphs are used to represent note duration.
4. Gridlines are used for both the pitch axis and the time axis (for the latter, they divide the time into measures).
5. Special glyphs are used to represent missing values (pauses) for each voice.

The basic structure of the figure is a 2-D system with the time sequence running horizontally and a categorical axis running vertically. This simple system can be used to depict many forms of sequence data. Running along the footer of this chapter is an application to human genome data. The data consist of a strand of human DNA, a genetic sequence consisting of a sequence of nucleotide bases – adenine, cytosine, guanine, thymine – encoded by their letters A, C, G, T, and the basic form of this view is a plot of *sequence* by *nucleotide*, split across the pages. The visualization has been augmented by using color to show coding *exons* within genes – sections of genes that will get encoded and sent out of the nucleus. Since the full sequence is over 3 billion nucleotides long, and each page shows 500 nucleotides, we would need a 6-million-page book to use this technique comprehensively – given the author's rate of writing, that would not be available until around the year 23900, so only a small fraction is given here.

Figure 5.11 provides an alternative reworking of the sequence data. To conserve space, we return to a 1-D representation, instead using color to encode the nucleotide and then glyph shape for the exon. Although we could fit more data in, and see it more clearly (it would be possible to click on individual nucleotides in this view; in the musiclike view running in the page footers, it would be much harder), the patterns are harder to spot – we are not as good at perceiving patterns in colors as we are spatial patterns. This figure serves a different goal – it emphasizes sequence

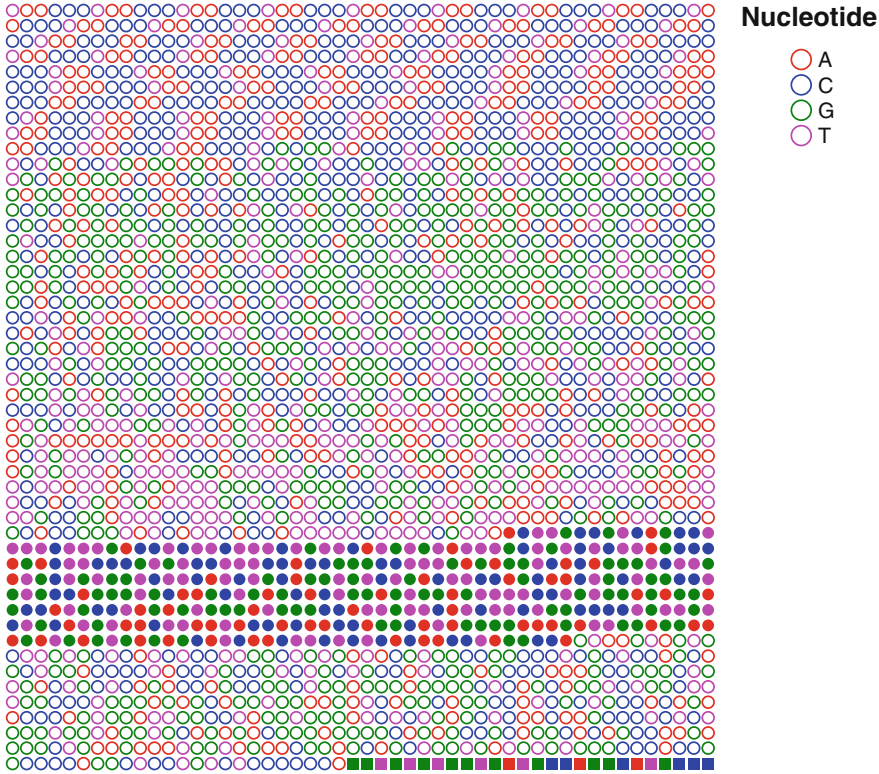


Fig. 5.11 Human genome sequence. 2500 nucleotides from human DNA, wrapped into a square. Color encodes the nucleotide, and shape is used for exons; *filled shapes* are exons, with different symbols for each different exon; *open circles* are nonexon nucleotides

patterns less and overall distributions more. Even though Fig. 5.11 is ideal for this technique – we have a small number (four) of categories, making the mapping to color work well – the chart is less useful and should be thought of more as a fallback design when *p*osition cannot be used.

5.3.1 Patterns in Sequences

In both the nucleotide data and the musical data, there is a common desire – to see patterns within the sequences. When using aesthetics to encode the variable of interest within a visualization of a sequence, it can be hard to see deep structure and mentally correlate patterns, especially when patterns occur on multiple levels. Instead of enhancing the elements with aesthetics to show a category, an alternative is to show the correlations between patterns directly.

Fig. 5.12 Goldberg Variations as shown by the *Shape Of Song* technique, courtesy of Martin Wattenberg. The notes of the piece lie along the horizontal axes. Identical sections are linked using an arc. One immediate feature is that the piece divides into four quarters, with the first and second quarters being identical and the third and fourth quarters also identical

Figure 5.12 shows one way of achieving this goal. This is a display invented by Martin Wattenberg [126], the goal of which is to explicitly show patterns in the data. Wattenberg describes the technique:

> The diagrams in the Shape of Song display musical form as a sequence of translucent arches. Each arch connects two repeated, identical passages of a composition. By using repeated passages as signposts, the diagram illustrates the deep structure of the composition.

The arcs drawn comprise a new element for the chart. They are *e*dge elements – more typically found linking nodes in a node-and-edge diagram or linking residuals to a fitted value. Rather than displaying the edge between two points as a simple line (which would cause strong overplotting issues), the edges are drawn as arcs so as to allow them to be more easily distinguished from each other. One reason this display technique works well is that visually significant edges correspond to repeated patterns that are *long* and *distant*. Long repeated patterns are more interesting than short ones, and patterns that return after a gap in the music are also more interesting from a musical point of view.

Figure 5.13 shows the use of this technique in a different context: spotting patterns within a sequence of genetic material taken from the human genome. The data format is the same as in the music example and the presentation technique is similar, but the details lead to distinct differences. One detail that makes a big difference for the usability of the result is the number of different possibilities for the categories. In music, notes have a fairly wide range of pitches, which, when combined with note lengths, makes for a large number of possible categories for each note. In our genome data, by contrast, there are exactly four categories: $\{A, C, G, T\}$. This means that for the music data, a sequence of three identical notes is quite important – if we played notes randomly the chances that three will match any other given three notes is low (back-of-the-envelope calculations make it about a one in 10,000 chance), whereas for nucleotides it is only one in eight, and we will see a lot of those connections in a 1000-nucleotide sequence, even by chance. Because this chart depends for its usability on data properties, this is a *fragile* technique, one that will usually require the user to modify a chart for different data sets.

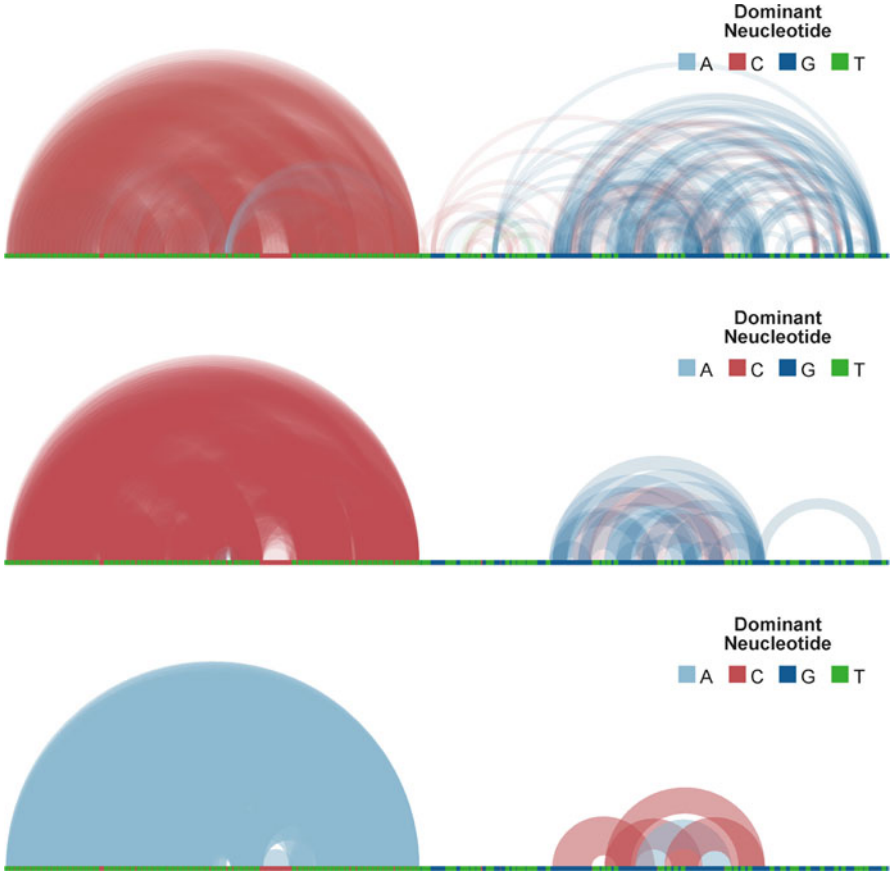


Fig. 5.13 Human genome data. Principles similar to those used by Wattenberg for songs are used here to link similar runs of nucleotides within a short section of 1000 nucleotides. Edges are created that link together runs within the genetic strand that have identical nucleotide patterns. Rather than display all such runs, a parameter that filters out smaller runs has been added to the algorithm. The *top display* shows only links for which $min(length) \geq 6$, the *middle* one links for which $min(length) \geq 12$, and the *lower* one links for which $min(length) \geq 24$. The transparency of the links has also been adjusted for each figure, so that figures with more links have a higher transparency. The color of the edges denotes the most common nucleotide in the pattern (using the mode statistic). The data are the same DNA data as shown in the running footer in this and in the following chapter

Fortunately, we can work around this issue. In Fig. 5.13 we have used two techniques:

1. **Filtering**: The three displays filter out shorter sequences so that only more important sequences are shown.

2. **Mitigating overplotting**: In each chart the level of transparency of the edges has been set to produce a more interpretable view.[4]

In this example, the two "tuning parameters" were chosen by hand. However, it is possible to derive reasonable algorithms to determine the two values. For example, if we assume the data given are randomly ordered, we could work out mathematically the minimal sequence length that should be shown that will be significant at some level[5] and use that as our filter criterion. On the other hand, the value of a visual technique is that optimal views are not as necessary when it is relatively inexpensive to show a few variations. From Fig. 5.13 we can see the main effects clearly: The first section of DNA contains much repetition, with the latter half having some self-similarity that appears more complex than the first half.

5.4 Summary

Mapping data to position is the best way to allow judgements to be made about those data. Since time data (but not elapsed time) are almost always never thought of as being predicted or influenced by another variable, but is usually thought to influence or predict other variables, the default assumption should be that we want to place time within the chart as a position, and preferably on the horizontal axis. For event data, where the data are simply point events in time, a 1-D chart may be no more than this single positional element, but this is a rare simplification, and this chapter shows how to expand and build on the basic concept with jittering, dodging, stacking, smoothing, and linking. Special techniques for regularly spaced "sequence" data have also been given. The overall goal is to take a strong core technique – time on the horizontal axis – and enhance it for a specific task. Given the power of the basic method, it is hard to go too far wrong with a design based on it.

5.5 Further Exploration

Robbins [92] provides many examples of 1-D charts (or charts containing a strong 1-D component) and good design suggestions for them. Cleveland provides a good discussion of dot plots in [24]; Tukey and Tukey suggest an interesting modification on simple jittering for 1-D charts in [113].

[4]This is a simple and effective general technique for modifying plots that have significant overplotting. The degree of transparency is a tunable parameter. If the elements are relatively opaque, then single items will be clearer and so outliers and unusual values will be more distinct, but differences in dense areas will be hard to perceive. Conversely, high transparency makes single values harder to observe but aids comparison of densely overplotted regions.

[5]It might form an interesting problem for statistical students ...

Chapter 6
Coordinate Systems, Transformations, Faceting, and Axes

If you're lost you can look - and you will find me
Time after time
If you fall I will catch you - I'll be waiting
Time after time

— Cyndi Lauper, *She's So Unusual* (1984)

6.1 Time Series

The previous chapter previewed the use of 2-D coordinate systems, showing examples of 2-D charts – `time` by `category`. When we have a continuous variable that changes over time, by far the most natural way to display it is as a *t*ime series chart: a chart with `time` on the *x* dimension, and the variable of interest on the *y* dimension, represented by a *l*ine element. Many examples of this chart are shown throughout the book, and the basic form needs little explanation. However, there are a number of details that can enhance this already thoroughly useful visualization.

One consideration is the choice of element to use. We looked at element choice in Sect. 2.2 on page 23. That section discussed whether a line or an area was more appropriate for a set of data, and one additional point to consider when making that choice is whether you want to stack the variable in the *y* dimension. An equivalent question is: Can the quantity be split by groups into subquantities that sum to the overall quantity? In the 1-D case, Fig. 5.9 shows an example where stacking works, since counts are stackable. When we have a variable assigned to the *y* dimension, we have to be more careful – sums are safe, percentages are if they have the same base, and means and other averages are not. Figure 6.1 shows an example where we stack sums. Using lines instead of areas would not make the stacked nature as clear, and so the area is to be preferred.

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_6,

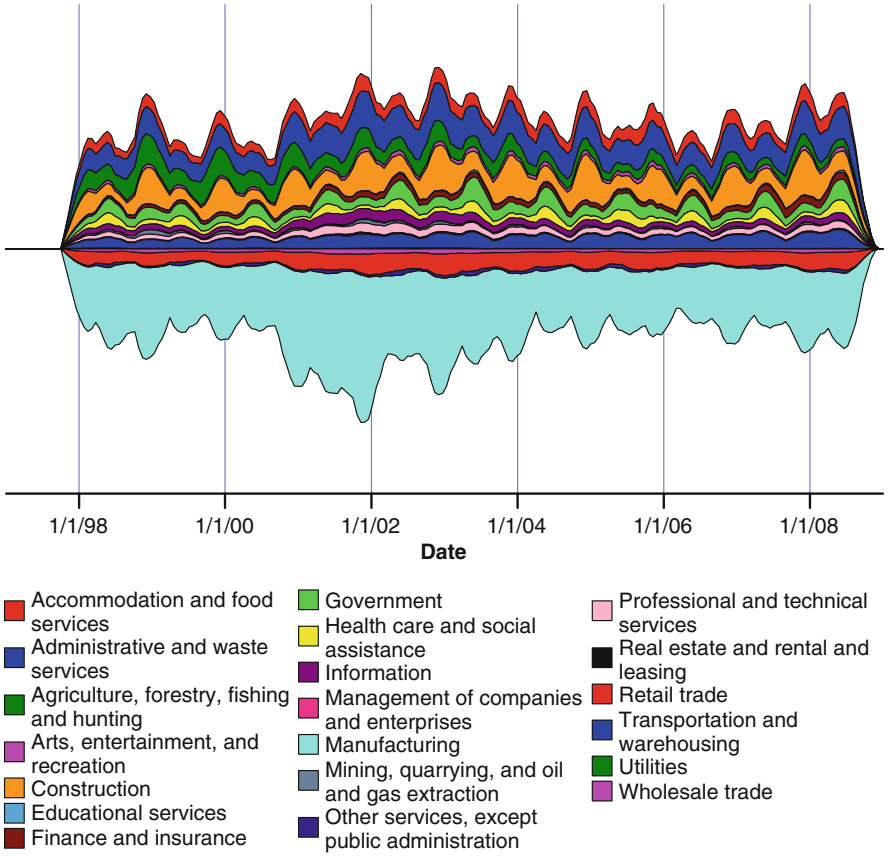


Fig. 6.1 Mass layoffs in the USA. A "mass layoff" is an event defined by the Bureau of Labor Statistics. It occurs when a company lays off a significant number of employees. This chart shows a center-stacked area chart where the height of the areas corresponds to the sum of the layoffs for a given month. The seasonality is clear, as is the huge importance of the `manufacturing` segment. This chart is constructed using the same formulation as the "ThemeRiver" chart defined in the previous chapter, but the difference in data makes it visually quite different

6.1.1 Aspect Ratio

A second choice, often overlooked in the creation of all types of 2-D charts, is the choice of the aspect ratio of a chart – the ratio between the physical extent of the x and y dimensions. Cleveland [21] introduced the concept for time series graphs and argued that since it is easiest to distinguish differences in angles if they are around 45°, then the optimal aspect ratio for a time series chart is one in which the

average absolute orientation of line segments in the chart is equal to 45°. Cleveland developed this idea in [22] and arrived at a suggested aspect ratio for which

$$\frac{\sum_i |\theta_i| L_i}{\sum_i L_i} = 45^\circ, \tag{6.1}$$

where θ_i is the absolute angle of each line segment in the line chart (a value between 0° and 90°) and L_i is the length of the line segment. Cleveland claims that this results in better looking charts than his original formulation, which did not weight by line segment length. As well as making an aesthetic improvement, this formulation has a stronger theoretical justification: It makes the measure applicable to charts where the line segments are an approximation to a continuous line – not just line segments based on raw data.

For example, consider a smooth (a spline smooth, a loess smooth, exponential smoothing, or some form of kernel smoothing) applied to some data. Theoretically the line will be infinitely smooth, and so the equation above, which when the line segments are infinitely small becomes a well-defined integral, is valid. More importantly, an implementation of a smoothing algorithm might produce line segments of varying lengths – in regions of more rapid change, more segments may be produced – and so weighting by the segment sizes reduces the dependence on the way the algorithm creates segments.

Further enhancements have been suggested for this approach, most notably by Heer and Agrawala [55], who provide alternative formulations and suggest approaches that can show aspects suitable for multiple scales of effect (both high-frequency and low-frequency effects). However valuable these techniques, the first step of paying any attention to the aspect ratio is the most important, and we have simply used Eq. 6.1 in our work.

Rather than iterate to a solution for the optimal aspect ratio, which would be the best solution for an automated system, a simpler suggestion is simply to consider the resulting aspect ratio as a diagnostic. In a graphics pipeline system, it is not a hard task to add a step toward the end that simply calculates the left-hand side of Eq. 6.1 and presents it to the user as a diagnostic. Figure 6.2 shows an example of the diagnostic in action for some economic data on US unemployment rates.

In Sect. 2.2.3 on the page 28 some reasons were given as to why including zero on an axis as a matter of course is a poor practice. This discussion on aspect ratios provides another reason; if we include zero on a vertical axis where it would not otherwise be included, it will cause the line segments to become flatter. This may be helpful, making the chart fit Eq. 6.1 more closely, or, as shown in Fig. 6.3, it may produce undesirable results. For the data under consideration, the rates of change are of greater interest than the absolute values, and so a chart that does not include zero on the axis and displays the changes more clearly is to be preferred. An alternative would be to plot differences between values directly in another view, but that would obscure the longer-term trends, so if only one chart is desired, it is probably not a good idea to show a chart only of the differences.

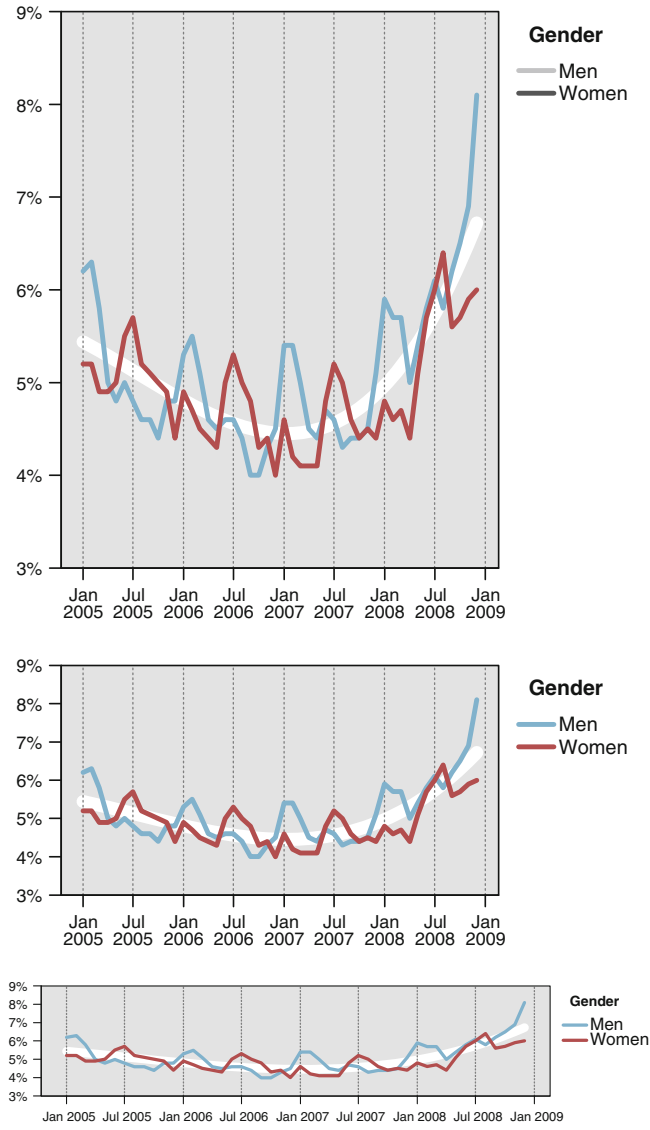


Fig. 6.2 US unemployment rates. The data consist of unadjusted overall US unemployment rates as reported by the Bureau of Labor Statistics. The rates are reported monthly for the years 2005–2008 and have not been adjusted for seasonality. Rates for men and women have been plotted separately, and a cubic polynomial fit to the overall rate is shown in *white* behind the series. The three charts are identical except for their aspect ratios. The average angles of the lines depicted are 67°, 45°, and 24°

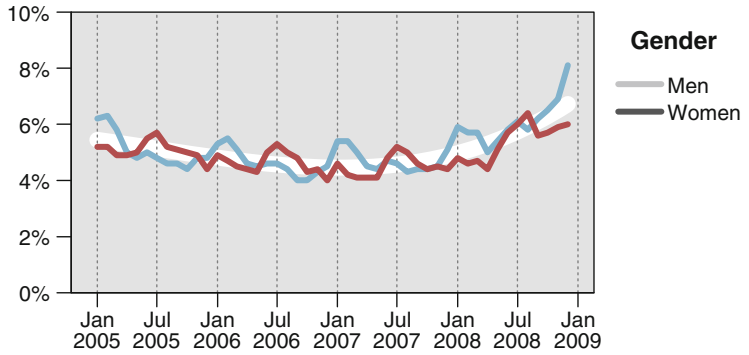


Fig. 6.3 US unemployment rates. This figure is similar to the previous figure showing unemployment data for men and women between 2005 and 2009. The difference between this figure and the second panel of Fig. 6.2 is that in this figure we have forced the vertical axis to include zero

As an aside, Fig. 5.9 on page 115 has an average angle of 67° for the bounds of its areas, and the chart does indeed look "vertically squished." A version twice as wide fixes this problem, but it is too wide for the page, and a version half as high is too small to read the text. Since the names of the movies are critical to the goal of the chart, the version presented is the best compromise. Compromises are common in chart creation, and this again argues for the use of the aspect ratio as a diagnostic aid, rather than an absolute rule.

6.2 Coordinate Transformations

Many of our metaphors for time are spatial in nature. We say that events are "far away" or "distant," we can "look forward" to occurrences, and so on. It is natural for us to map time to a spatial dimension. So far this chapter has considered only linear mappings for time. Time, after all, "flies like an arrow" or "flows like a river." Some meandering may occur, but most of our metaphors portray time as a straight line. That is a good start, but there are cases where linearity can be limiting.

Taking our cue from the phrase "what goes around comes around," one important transformation is the *polar* transform. Much time series data are cyclical in nature and displays that highlight that periodicity are important. An analog clock shows the cyclical nature of time, with a 12 (or occasionally 24) hour period, and we can generalize that to other displays, such as that in Fig. 6.4 in which we display data on mass layoffs using a polar transformation that has a periodicity of 1 year. The single line that traces the number of events wraps around the chart several times, showing a consistent pattern with mass layoffs peaking near the end of the year, with a secondary peak in early summer.

Even when we only show data for one period, using a polar transformation can be revealing. In Fig. 6.5 a simple polar transformation has been applied to the movie

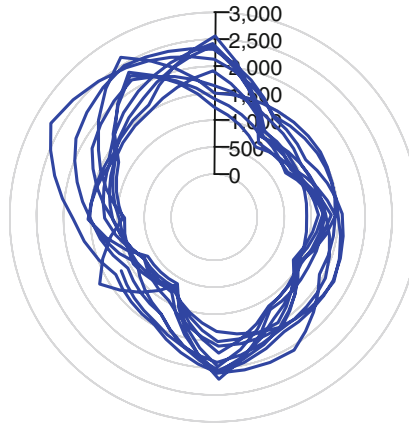


Fig. 6.4 Mass layoff data of Sect. 6.1 displayed using a cyclical polar transformation. The *line* shows the total number of mass layoff events on a monthly basis over the time period 1998–2008. The polar transformation is cyclical, with cycle length exactly equal to a year and runs clockwise

chart (Fig. 5.9), resulting in a chart that better shows the periodicity of the data. This version is more compact than the linear version, and it visually appears more like a single entity, which makes the chart easier to compare to other similar charts. For the movie data, we could categorize movies into types – drama, comedy, action, etc. – and produce small multiples of the basic chart for each category. This faceted version would allow us to compare patterns between movie genres and answer questions such as whether action movies show yearly patterns similar to those of comedies or whether horror films have higher peaks or longer tails than dramas.

Figure 6.6 shows a more unusual transformation – one that is, unfortunately, unlikely to be readily available in most software packages. The transformation is defined as

$$(r, \theta) = (\sqrt{t}, 4\pi t).$$

The value 4 in the equation indicates that the spiral will wrap around the circle twice ($2\pi t$ would wrap once). This transformation takes the 1-D time axis and places it within a 2-D coordinate system. Although it is probably not available as a built-in function, this transformation can be simulated by creating derived variables for r and θ and then using those variables.

Using derived variables (or *formulas* in Excel-speak), it is possible to simulate a coordinate system by transforming the data before plotting and then displaying the transformed data. This will work subject to the following constraints:

- The data must be plotted without any statistics that modify the values since the values are no longer "real" but are simply precalculated coordinates.
- No shapes will respect the coordinate system. For example, a bar chart transformed into a polar coordinate system in this way would not have bars that look like sections of a circle. It is best to stick with points (which have no size and therefore do not need their shapes transformed) and line segments, with the understanding that the line segments will simply connect two (transformed) points and not be transformed into paths that respect the coordinate transformation.

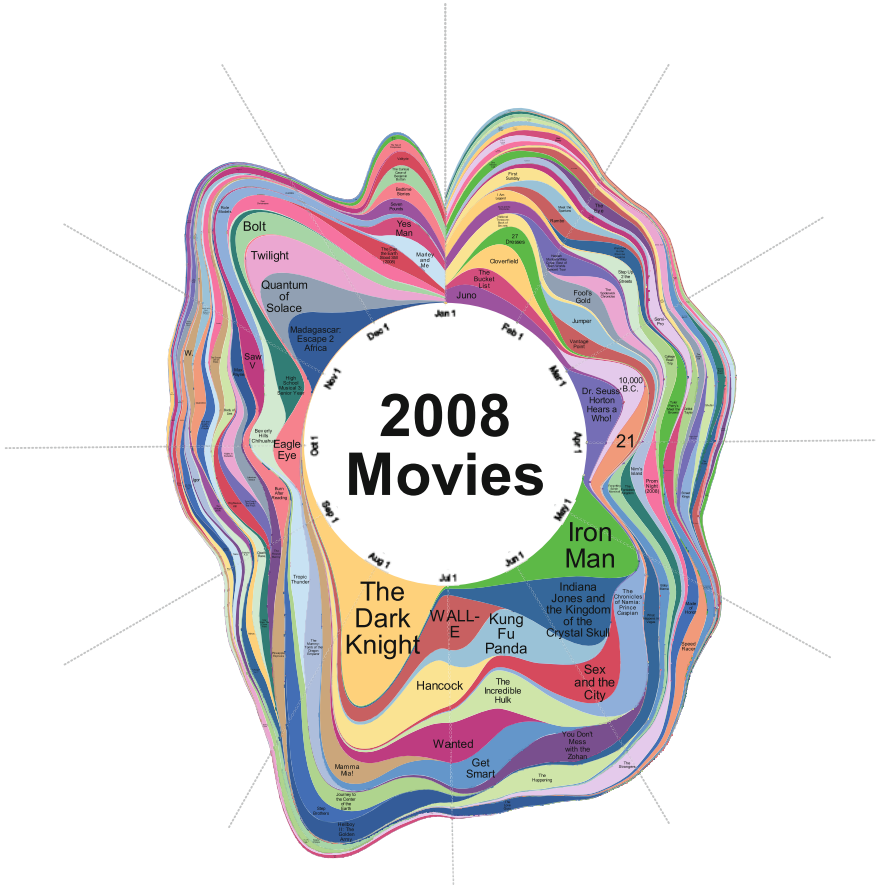


Fig. 6.5 US box-office attendance for movies released in 2008. This figure is identical to Fig. 5.9 on page 115, except that we have added a polar transformation, and instead of using the special *c*enterStacking position modifier, we have used the simple stack position modifier discussed in Sect. 5.2. This display emphasizes the yearly movie cycle; both the summer blockbuster and the Thanksgiving release time frames are evident. Note that there is no edge correction in the density estimate, and so the densities around the beginning and end of the year cannot be taken as accurate

- Axes will not be useful as they will show the transformed values, which are of no interest to the reader, rather than showing guides for the original data.

6.3 Axes

Given that we have a chart, using a coordinate system and possibly some form of transformation, the mapping from data to locations may not be clear. That is where axes come into play. An *axis* is a visual representation of a dimension of a coordinate

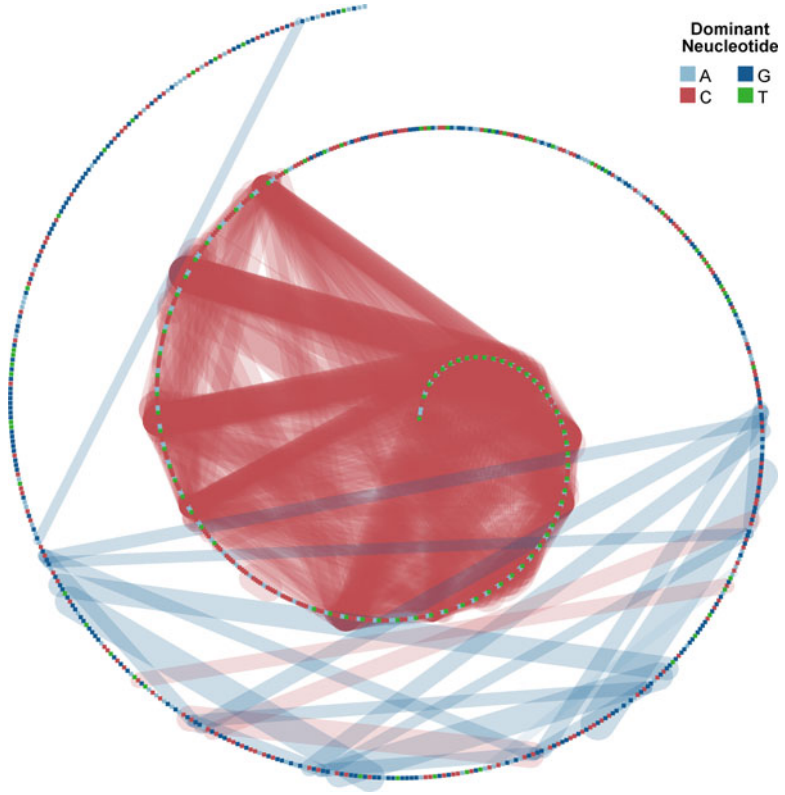


Fig. 6.6 Human genome data. Figure 5.13 on page 120 showed the relationships between patterns of runs of nucleotides for some human genome data. In that figure the data were laid out on a *straight line* and the links between similar sections shown as *semicircles* so as to avoid all the links obscuring each other in one dimension. In this figure, we have adopted a different approach. The data are laid out in a spiral, and this allows the links to be drawn as *straight lines*

system. Wilkinson [135] defines axes as legends for a position – a similar statement. The core concept is that an axis should provide a guide as to how the mapping from data to screen (or paper) location occurred. In this section, I consider gridlines, ticks, labels, and titles all to be part of an axis. One point to note is that other visual representations of a coordinate mapping are possible – axes are not the only way to do so. Other guide techniques include showing transformed unit vectors in each dimension, such as in Fig. 6.7. This display of small vectors to show an underlying field is probably most common to readers when used to show wind or ocean currents on a map, where each point on the map has an associated vector with both direction and magnitude. The difference here is that we do not have a field defined on a flat space; the space itself is distorted, and the vectors show that distortion.

One of the first things to ask about an axis is whether or not one is necessary in the first place. Tufte [111] asks us to minimize the amount of nondata ink in the

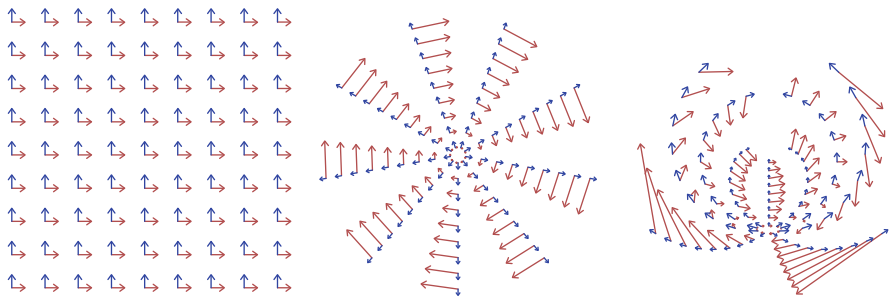


Fig. 6.7 Unit vectors as coordinate guides. At each grid location in the base a 2-D coordinate system (for an arbitrary grid – a 9×9 grid in this example), two vectors have been drawn in the x and y dimensions. In the *left* panel, there is no coordinate transformation. In the *middle* panel, a polar transformation has been applied. The sizes of the *red arrows* clearly indicate how a fixed difference in the angular direction leads to very different sizes when the radial coordinate changes. It also shows that the dimensions are still perpendicular to each other. In the *right* panel, a polar transformation has been applied a second time, leading to a coordinate system that is highly distorted and not orthogonal. This is the "bad" coordinate system shown previously in Fig. 2.27 on page 53

chart, and all authors agree that the most apparent feature of a visualization should be the data. With this in mind, the default position should be not to include an axis. And yet most charts shown *do* include axes – in fact it is typically the default setting for any charting system. The answer is that, most of the time, there is indeed a compelling reason to show axes.

An axis (and its dependent parts) allows the user to look at a chart feature and get a fairly precise data value indicating the "location" of the feature. This "location" is in data coordinates, so for simple charts, it allows us to read off the data values. Often this is of value. For time data in particular, drawing an axis for time is almost always a good idea since if a chart shows a feature of interest, you can be fairly sure that knowing when it occurred will be important to the reader of a chart. Even in Fig. 6.5 on page 129, in which a good axis is hard to draw due to the transformation, we have at least shown tick marks for the months to make it clear when summer, Christmas, and Thanksgiving holidays occur. When this chart was created, it was clear that knowing when movies were released would be of strong interest. Hence the need for an axis for time.

For the same figure, there is no axis for the other dimension; that dimension shows the number of people attending movies. Although that information could be of interest, the main goal of this visualization is to show the *relative* sizes of the audiences. The fact that a certain number of people attended movies in late July is of less importance than the facts that the peak occurs in late July and the number of people attending is two to three times larger than the numbers attending in subsequent months. Adding an axis would add detail that reveals nothing interesting about the data.

Leaping ahead somewhat, Fig. 6.13 on page 140 gives another situation in which axes are superfluous – when we are interested in *comparing distributions*. The interesting features of this faceted chart are the differences between the temporal distributions; some days have a certain pattern, other days have a different pattern. The caption of the figure gives the range of the time dimension, and so at least some context is preserved. A valid argument might be made that we should show the time axis in Fig. 6.13. There are some interesting peaks and valleys in the data – when do they occur? This leads to another consideration in deciding whether to show an axis; has that information already been given in another chart? For this example the answer is yes. In Fig. 6.12 on page 139 the peaks and troughs are visible and the chart is decorated with axes for all dimensions. We can see the peaks in travel at 1300 for both weekdays and weekends, and the big peak around 1830 for weekdays. Axes in the subsequent figure are not as necessary, since the information has already been given out. Repeating it would be unnecessary. It serves no purpose to say the same thing another time.

Overall, the goal of an axis is to allow the reader to do a reverse mapping from the visual space of the chart to the data values. When that mapping is not necessary, when it would distract from showing more important features of the data, or when there simply isn't enough space to show a good axis, then do not show the axis. If it reveals data or would help a reader apply exterior knowledge (such as "people go to movies in the summer"), then an axis is valuable and should be included.

6.3.1 Drawing Time Axes

Drawing an axis is a tricky proposition, even for a simple axis on an untransformed dimension. A general algorithm for drawing an axis might be as follows:

1. Calculate the range that the dimension spans. This could be the range of the data, the range of the output of a statistic being calculated on the dimension, a range based on theory (such as $[0,1]$ for a probability), or a user-defined range.
2. Calculate a delta value σ for the distance between ticks on the axis. This is a value that makes sense to someone viewing the charts as a good value by which to separate ticks.
3. Optionally, use σ (or a fixed small margin) to expand the range to cover a larger area. This can be done simply to prevent data items from running up against the edges of the chart area (or adjacent panels in a faceted chart) or to ensure that the axes span a pleasantly readable range ($[0,200]$ is easier to work with than $[13,191]$, for example).
4. Pick a value α on the axis at which to place a tick (for numeric data, values of $\alpha = 0 + n\sigma$ for $n \in \{0,1,\ldots\}$ work well in most cases).
5. Draw ticks at all points on the axis offset from the start value by multiples of σ.
6. Divide up each gap between major ticks into η divisions and draw minor ticks at each division point.

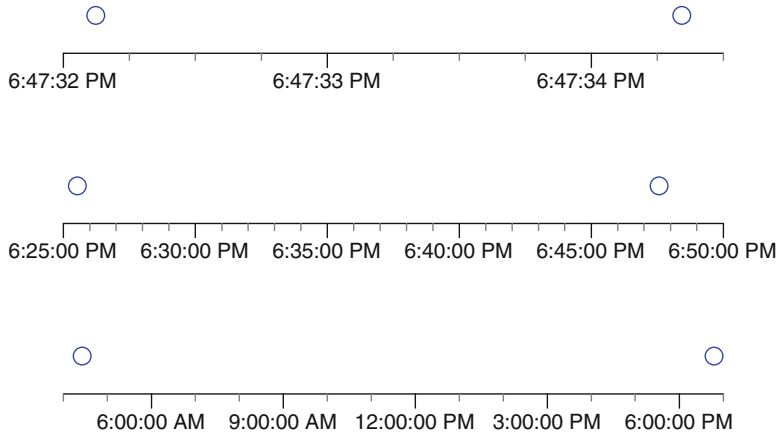


Fig. 6.8 Choosing (α, σ, η) for short time durations. These charts show default choices of axis parameters for time spans consisting of a few seconds (*top*), minutes (*middle*), and hours (*bottom*). The data (shown as *points*) are simply two values used to define a suitable range. Note that in these charts we are using an algorithm for step 3 that expands the ranges to the next major tick unless that creates too big a margin, in which case we expand to the next minor tick

Each of these steps requires a fair amount of work, and visualization experts will often disagree over details. In step 1, should a dimension for a smoothing statistic have a range of just the smooth values or the whole data? In step 3, is it better to expand the dimension to nice values or just expand by a small fixed amount? Should the axes be drawn all the way to the expanded ranges or just to the ends of the data? In step 6, should we draw minor ticks on the edges past the first and last major ticks?

A complete system will give you control over all of these possibilities, but the defaults must be good. The goal of a guide is to help you understand the data, and a guide that needs a lot of guidance to function correctly is of little value.

For time data in particular, the choices of (α, δ, η) are critical to creating a useful chart. Unlike most numeric data, time does not have a standard "zero point" or origin that we can use to base ticks from. Also, time is measured in units that are not only nonmetric (so simple rules like using multiples of 10 are useless), but the units themselves are variable – one month is a unit that is not always of the same duration!

Figure 6.8 shows a set of default axes for data of short time spans (less than a day). In these figures, the first choice to be made is which time unit is going to be used to define the ticks. A good general rule for axes is that five to nine ticks make a suitable axis, so a simple rule is: Choose the unit that divides the chart into a number of chunks that is closest to seven. In Fig. 6.8a, dividing by seconds into three parts is preferable to dividing by minutes into one, so we use *seconds* as our unit. For Fig. 6.8b, 24-*minute* divisions makes a better choice than one-*hour* divisions or thousands of *second* divisions, and in Fig. 6.8c, *hours* are the best unit.

With the units decided, the next step is to work out what multiples of the unit to use. Each unit has a set of standard multiples that provide readable results; they are

Table 6.1 Multiples and divisions for common time units. For each time unit, the middle column lists suitable multiples that make sense for it. For example, showing multiples of 7 or 14 days is preferable to showing ticks every 10 days. The last column gives a suitable number of divisions for that unit to provide minor ticks, so days should be divided into four minor divisions, giving ticks every 6 hours

Unit	Possible multiples	Minor tick divisions
Second	$\{2,3,4,5,6,12,15,30\}$	5
Minute	$\{2,3,4,5,6,12,15,30\}$	4 (every 15 seconds)
Hour	$\{2,3,4,6,12\}$	4 (every 15 minutes)
Day	$\{2,7,14\}$	4 (every 6 hours)
Month	$\{2,3,4,6\}$	4 ($\approx$ every week)
Year	$\{2,4,5,10,20,100,200,500,1000\}$	4 (every 3 months)

given in Table 6.1. We try each of the multiples for our chosen unit, and the one that gives us closest to seven divisions is the one we choose.

For Fig. 6.8 we find the following results:

$$\sigma_1 = 1\ \textit{second},$$
$$\sigma_2 = 5\ \textit{minutes},$$
$$\sigma_3 = 3\ \textit{hours}.$$

To find a good value for the number of minor divisions, η, first consider the multiple we used when deciding the major divisions. If that multiple was a reasonable one (in the range $2\ldots9$), then use that number of divisions. If it is larger than 9, then divide it evenly to get closer to 3 to 7 minor ticks. For example, if the major tick division was 15 minutes, we might divide by 3 to choose five minor tick divisions. If the multiple is 1, use the value given in Table 6.1 for minor tick divisions.

Fortunately time data do not impose any difficulties for the calculation of α. We pick the low point of the range, zero out all time units up to our unit of choice, and use that value. This results in the following values that define the axis for Fig. 6.8.

$$(\alpha_1, \sigma_1, \eta_1) = (6:47:32{,}1\ \textit{second}, 4);$$
$$(\alpha_2, \sigma_2, \eta_2) = (6:25:00{,}5\ \textit{minutes}, 5);$$
$$(\alpha_3, \sigma_3, \eta_3) = (3:00:00{,}3\ \textit{hours}, 5).$$

Figure 6.9 shows which axes have extents longer than a day. For these sample axes, the following parameters can be derived using the algorithm described above:

$$(\alpha_1, \sigma_1, \eta_1) = (\textit{September}\ 13{,}7\ \textit{days}, 7);$$
$$(\alpha_2, \sigma_2, \eta_2) = (\textit{February}\ 1{,}1\ \textit{month}, 4);$$
$$(\alpha_3, \sigma_3, \eta_3) = (\textit{January}\ 1992{,}2\ \textit{years}, 2).$$

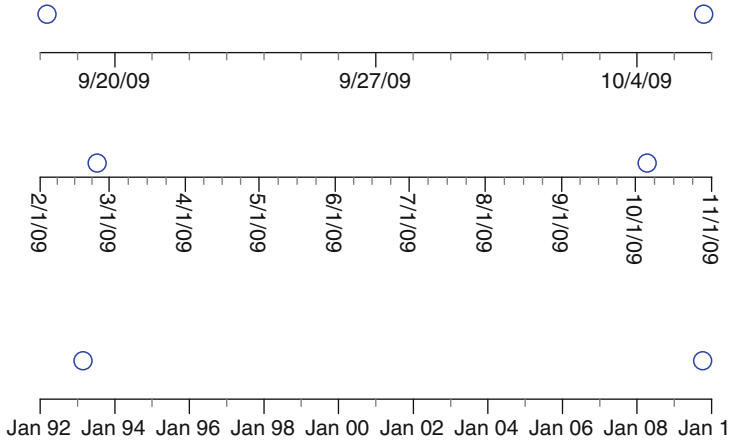


Fig. 6.9 Choosing (α, σ, η) for longer time durations. These charts show default choices of axis parameters for time spans consisting of a few days (*top*), months (*middle*), and years (*bottom*). The data, as in the previous figure, are simply two values used to define a suitable range

6.3.2 Formatting Time Ticks

Formatting time values is another hard problem. There are two approaches commonly used. One is to ask the user to input a format as a coded string; the second is to pick a suitable choice using the known locale the program is working in. The *locale* is a combination of the country, language, and other information needed to give reasonable formats for numbers, time, and currency. A quick look at a program like Excel shows both methods in operation; a default locale-dependent format can be used for time values, or a fixed pattern (such as "mmm:YY" indicating a named month separated by a colon from a two-digit year).

It would be nice to be able to specify something a little more general, like "show month as a name and year," but allow the program to format those elements with respect to the known locale. Few systems currently allow such flexibility at present, however. As mentioned above, good defaults and the ability to customize fully, in all locales, is a necessity.[1] Figure 6.10 shows some variations in formatting for the simple axes shown in this section, with different locales for Fig. 6.10b and d. Good labels, correctly localized, make charts much more informative. Your guide should not only speak your language, but he should speak it the way you want.

[1]The chart-drawing package designed by the author has been in use since 2002 in over 150 countries by a large number of people from different disciplines. Feature requests for formats like "how do I specify a format to show the *N*th week of the year" or "how do I get an axis with values 1Q02, 2Q02, ... to work in Japan" and bugs like "8.08 is not being recognized as a date in Germany" are common and indicative of the importance of getting the labels right.

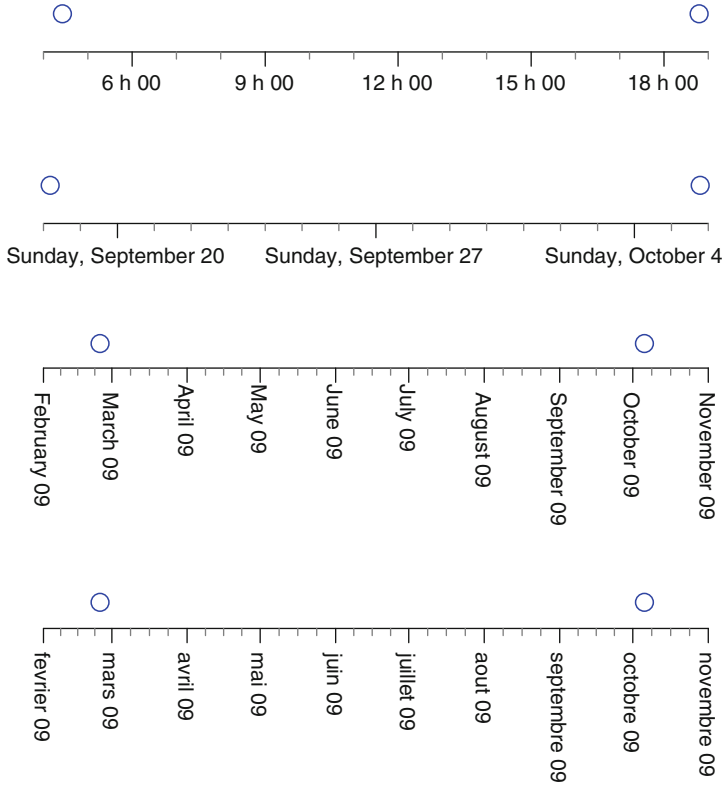


Fig. 6.10 Tick label variants. These four figures show axis labeling variations based on Figs. 6.8 and 6.9. *Top to bottom*: Standard time format, showing only hours and minutes, in a French locale; day of week, day of month, and month name in an English locale; month name and two-digit year, padded with zeros, in the USA; month name and two-digit year, padded with zeros, in France

6.4 Faceting

Faceting is a simple way of making more use of a chart that is already good. If we have an informative chart that gives us useful information, but we have other variables that we would also like to consider, then faceting is a way of adding those variables to a chart without modifying the basic utility of the chart. In its simplest form, faceting creates multiple charts using the same data but splitting the data between the charts according to a categorical variable. As an example, consider Fig. 6.11.

This figure is based on a data set consisting of every play made in every game played by the Chicago White Sox, a baseball team that plays in the American

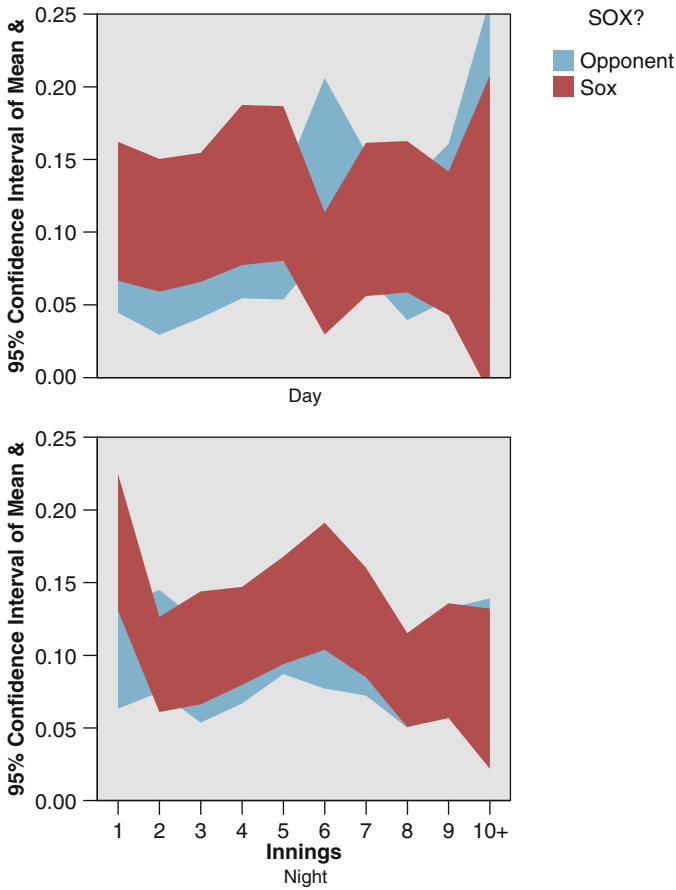


Fig. 6.11 Average number of runs scored for each inning in the 2005 White Sox season, shown by a confidence interval for the mean. The chart has been faceted by the binary *DayNight* variable to show the difference between day and night games. There is some evidence for two interesting effects; in night games the Sox often get off to a good start in the early innings, as compared to the other team and as compared to their performance in day games. Second, there seems to be an interesting difference for them in the sixth inning between day and night games

League, for the year 2005, the year in which they won the World Series.[2] In this figure we are looking to see if the number of runs expected to be scored has any relationship to the innings. One might theorize that pitchers or batters take time to settle in, or conversely get tired, and so some effect might be evident. To test this, the

[2]The World Series in baseball is a competition restricted to teams playing in the major leagues in the USA. There are two different leagues, and there is a significant rivalry between the Chicago Cubs, in the National League, and the Chicago White Sox, in the American League. For Cub fans, never fear – our turn will come, any century now.

base chart shown is a 95% confidence interval for the mean number of runs scored for each play. If we simply calculate the overall average, it is about 0.1 runs/play. We then start conditioning these data by splitting them into partitions, as follows:

1. The data are divided up into categories on the *x* axis using *innings*. This partitioning is shown on the plot by putting the innings on the *x* dimension (time running horizontally) and the confidence interval for the expected score is shown with an area envelope. At this point we have conditioned the data by innings and shown each inning as a vertical range for the confidence interval. The element chosen is an area element, but we could equally well have chosen interval bars.
2. Next, we split the area chart into two partitions, one for the White Sox' plays and one for the opposition's. We use the color aesthetic to differentiate the two area plots. The data are now conditioned on *innings* × *team*.
3. Finally, we split the data again into two groups, this time by whether the game was a day game or a night game. We use faceting this time, with a complete chart for each of the two types of games. The data are now conditioned on *innings* × *team* × em dayNight.

This example illustrates a common paradigm for exploring data. Start with a statistic or summary of the data of interest and then divide those data up into partitions using other variables. It may be strange to think of something as simple as a bar chart of counts in that way, but it helps in understanding the purpose of the chart and why it works: The fundamental measure is the count of how many data items there are. It is then split up by a categorical variable to show how many data items there are, conditional on each distinct value of that variable, and, equivalently, how many data items have that value. We are then faced with the challenge of how to show that partition. In the previous chapter we discussed the use of a *d*imension and indicated that it is the clearest way to show a variable, so usually we will want that to be the first choice. In Chap. 7 we will talk about using aesthetics, and in this section we explore faceting in more detail. Mathematically the effect will be the same. Visually they are very different.

6.4.1 Faceting by Time

Given that putting time on the *x* dimensions seems so obvious, why would we want to bother discussing faceting[3] by time? One possibility is that other variables are more important than time; another is that we have several scales of time and want to see multiple levels on the same chart. Figure 6.12 shows a set of data on passenger

[3]Although both "faceting" and "facetting" are valid spellings, the former seems more commonly used, as a quick Google search will indicate.

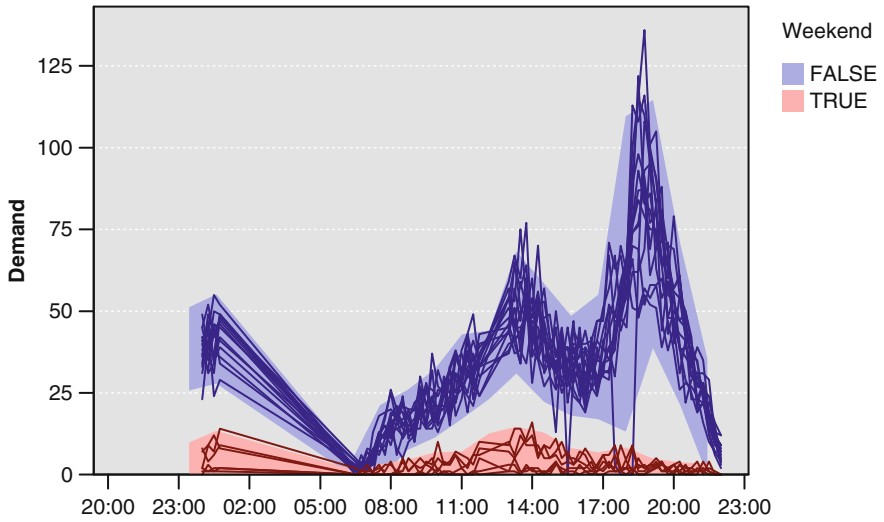


Fig. 6.12 Passenger arrivals at a subway bus terminal in Santiago de Chile for each 15-minute interval between 06:30 and 22:00 hours from 1 to 21 March 2005. Time series for each day are overlaid on top of each other, with the color of each time series representing whether the day is a weekend or not. An envelope for each group represents the 95% confidence interval for an individual value, conditional on the 15-minute time of day

arrivals. This data set was part of a competition [90] that had a defined goal of predicting arrivals, as stated in the competition abstract:

> A public transportation company is expecting increasing demand for its services and is planning to acquire new buses and to extend its terminals. These investments require a reliable forecast of future demand which should be based on historic demand stored in the company's data warehouse. For each 15-minute interval between 06:30 hours and 22:00 hours the number of passengers arriving at the terminal has been recorded and stored. As a forecasting consultant you have been asked to forecast the number of passengers arriving at the terminal.

Overlaid time series are a form of faceting; each time series is its own chart, and the layout for each chart is trivial – they are placed on top of each other. Overlaying has one major advantage: It makes it easy to compare relative distributions. In Sect. 5.1 on page 105 we followed Cleveland [21] in stating that aligned identical scales allowed the easiest quantitative comparisons, so this is unsurprising. However, the disadvantage of overlaying is that it necessarily hides some data behind others. When there are only a few items being overlaid (as in Fig. 6.11), that is not a major issue, but for this chart we have many overlapping series. We can see the difference between weekdays and weekends clearly, but perhaps we are missing some details within weeks, or between Saturdays and Sundays. Figure 6.13 uses faceting to explore the possibility.

In Fig. 6.13 we have moved the time series into panels, one for each day of the study. We have also been careful to ensure that the layout of those panels makes

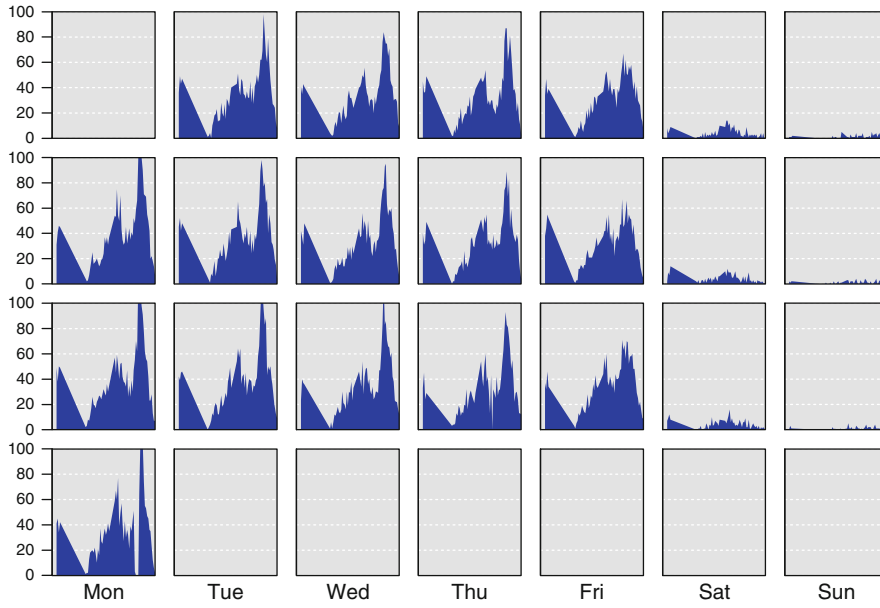


Fig. 6.13 Passenger arrivals at a subway bus terminal in Santiago de Chile for each 15-minute interval between 06:30 hours and 22:00 hours from 1 to 21 March 2005. Each day is placed in its own panel. Note that the area representation is a good one for this chart; for any range of times, the total area for that range is proportional to the total number of passengers demanding transport during that time

sense. In fact, this layout can be considered a *2-D faceting*. Horizontally the data are faceted by day of the week; vertically they are faceted by the week itself. The difference between weekdays and weekends is still clear, but we can also see now a difference between Saturday and Sunday, as well as a lack of difference between the weekdays. This *lack of difference* is important – we can now start formulating models in our minds. We can model the day-of-the-week effect by a categorical variable with three values – $\{weekday, saturday, sunday\}$ – and after creating that variable we can replace Fig. 6.13 with one using that derived variable for faceting. Such a chart would show a week with three panels – one with five overlaid time series for weekdays, one panel with a single series for Saturday, and one with a single series for Sunday. This visualization would be a strong graphical portrayal of the model.

6.4.2 Faceting Complexity

In Chap. 11 we will discuss chart complexity in more detail, but it is worth mentioning here because faceting is a powerful tool, and it is easy to get carried

away. A 3-D chart is more difficult to interpret than a 2-D chart, and 4-D charts require considerable mental energy to evaluate. When we add aesthetics, more than one or two and our perceptual cognition is overloaded with the task of sorting things out (for an example, refer back to Sect. 2.4.2 on page 46). Is the same true for faceting? What makes it too complex? Is it purely the number of variables, or is it the number of facets that are generated?

Unfortunately, there has not been enough research into this topic. There has been some advice on the presentation of tables, such as Miller [78], but this advice is more on how to lay out a table given that you want to show a number of variables rather than advice on how many is too many. If you have a number of variables (or *dimensions*, as they are often called in the world of tables), then suggestions for how to organize them include:

1. Respect logical groupings. If you have `weeks` and `days`, place `days` within `weeks`, rather than vice versa. Related items should be put on the same dimension whenever possible.
2. If there is no such grouping, place the dimensions with more categories outside and those with fewer categories inside.
3. Allocate the variables to rows and columns to match the desired aspect ratio of the chart as closely as possible.

The third item is one that has proved quite a significant one for the author, as highly faceted charts take up a fair amount of room and can be tricky to fit on a page.

Returning to the original question, I would suggest that the number of faceted items is not as important as the number of variables. Consider Figs. 6.14 and 6.15. Figure 6.14 shows a pie chart for each game played by the White Sox, with pie chart categories defined by type of play. Although there are 161 games displayed, the chart is not hard to understand; each chart can be compared to any other and the order in the list is easy to keep track of. In contrast, Fig. 6.15 is harder. It only contains 28 panels and has fewer symbols than the previous chart, but when comparing cells the reader is constantly required to refer back to the labels for the facets to work out what is going on. For this chart the strongest effect is the difference between the numbers of single plays that score one run in games won by the Sox as compared to games lost by them. The opponents, on the other hand, do not see the same relative increase in games they win. The inference for a sports fan is that it is the White Sox batting that determines the game – their pitching is not a major effect as the other teams score similar amounts in won or lost games; it is the batting that determines their success or failure.

This effect is tricky to find; the three dimensions of faceting make the chart challenging. As a rule of thumb, it is probably best to stick with one or two dimensions for faceting, unless the dimensions are strongly related to each other.

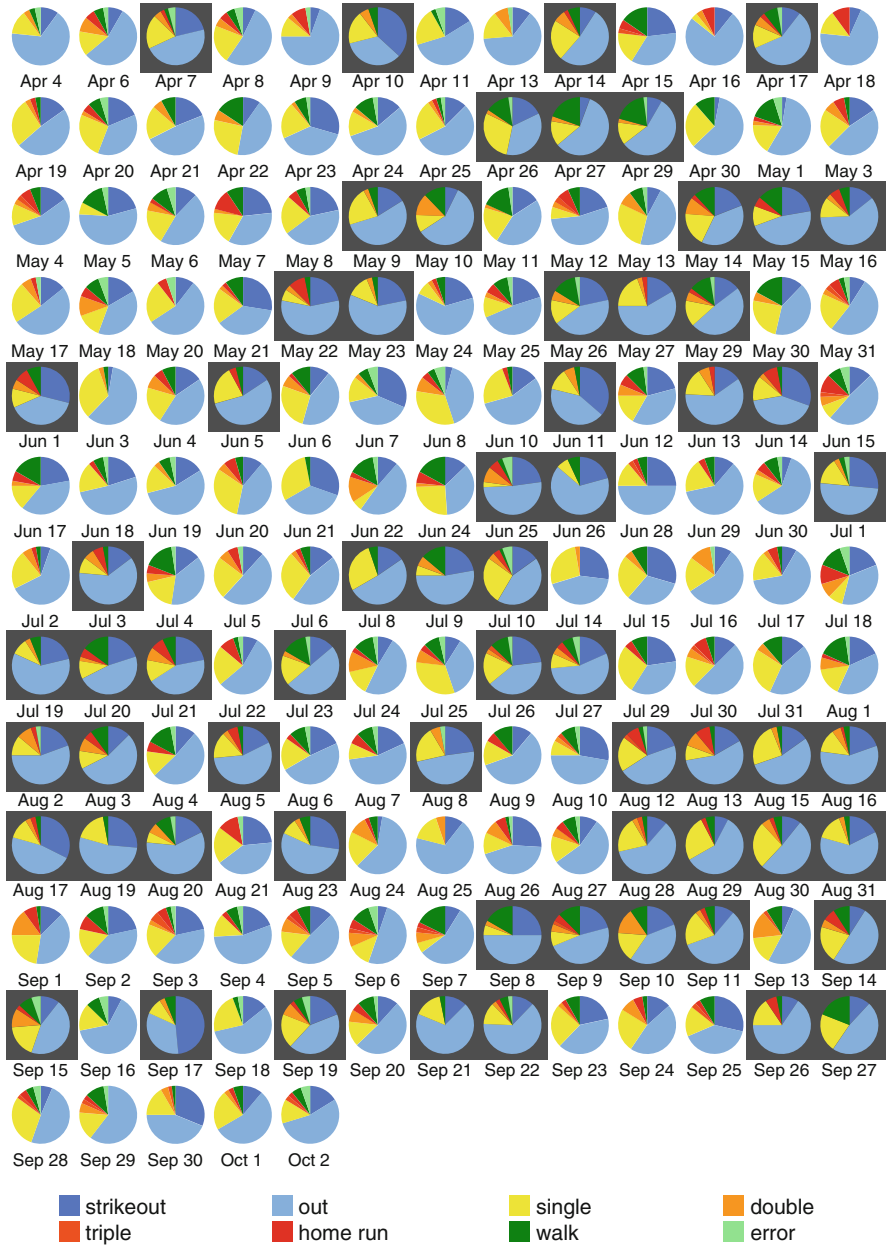


Fig. 6.14 White Sox plays made in the 2005 season. This figure shows pie charts indicating the frequencies of different types of play, with one pie chart for every game in the season. This was achieved by faceting the basic chart by the *date* variable. A *brightness aesthetic* has been applied to the cells that darkens the cell background for games that were lost

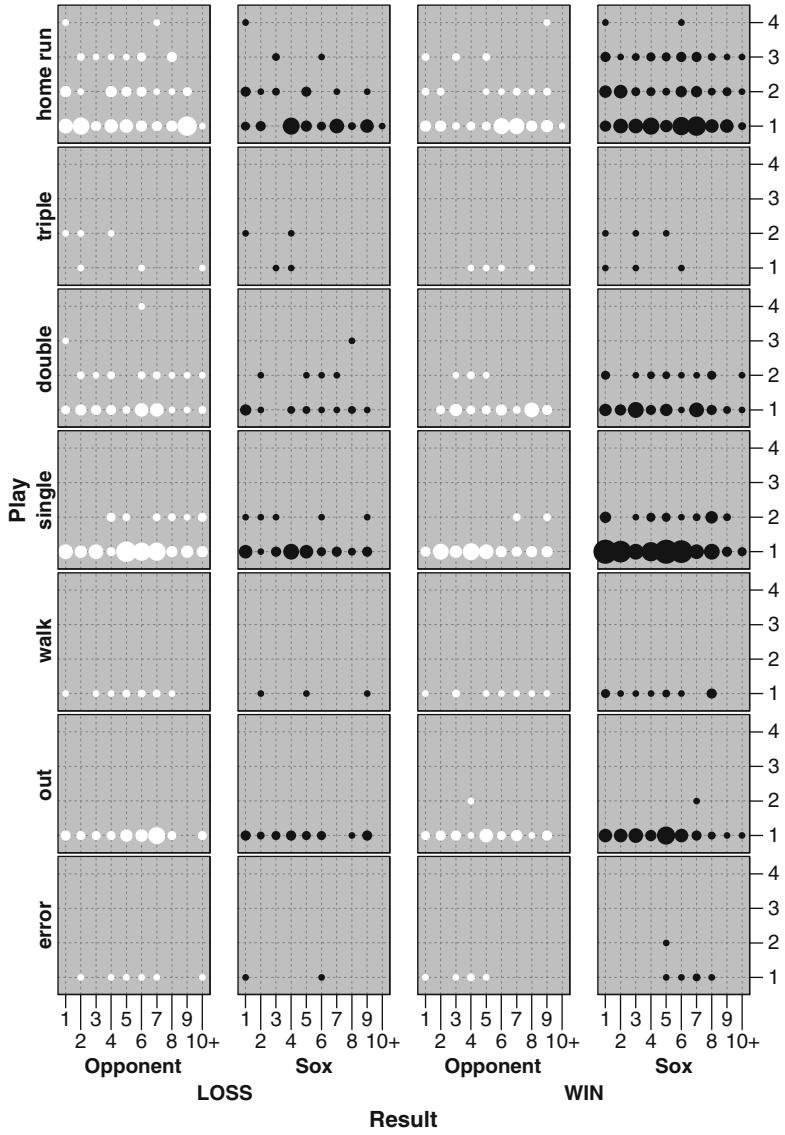


Fig. 6.15 The base plot in this display is a 2-D plot of innings against runs. Only innings in which runs were scored have been used for the visualization. The plot has been faceted vertically by type of play (filtering to show only the most common plays) and horizontally by team (Sox or opponents) and whether the innings were in a game that was won or lost

6.4.3 Time Within a Faceting

In the previous section we discussed the use of time as the faceting variable. The use of other nontemporal variables to facet a visualization of time data follows similar rules. The use of tabular layouts of time visualizations requires little additional advice; perhaps the only extra advice would be for 1-D facetings. In Fig. 6.11 on page 137 we used a vertical layout, with the facets drawn on top of each other. Since *aligned* comparisons are easier than unaligned ones, this makes the time dimension easier to compare than the response dimension (the mean `runs`). Placing the facets side by side would make the comparison of relative mean `runs` easier. Which is better depends on the goal of the visualization; is it more important to stress "when" or "how much"?

A paneled layout is not the only option, however. Wilkinson [135] discusses facet layouts (Sect. 11.3.5 of [135]) where the layouts can be considered as charts in their own right. The basic idea is that we take the faceting variables, make a graph out of them, and, instead of showing an element at each position calculated by that graph, show a different chart. This is a powerful technique as it allows two levels of layout, the outer faceting layout, and the inner coordinate layout. Wilkinson gives two examples. The first is a tree layout where the faceting layout is defined by a regression tree, and the inner chart shows the distribution of the predicted value in each node of the tree. The second example of note in that section is a polar layout, where a time variable, `month`, is used as the faceting variable, but the layout is circular rather than linear.

The scatterplot is the simplest 2-D faceting layout to use for this approach; and it has an important application in the area of spatiotemporal data. These are data where we have time-based information at a number of known geographic locations, for example, measurements of the amount of rain collected by month at a number of different weather stations.

The data for the following example consist of readings from a number of weather stations in the UK.[4] The data can be found at the UK Met Office site [48], and the map locations for the weather stations are also available on the same Web site.

[4]As a US and UK dual citizen of English birth, who has lived much of his life in Ireland, I appreciate that there might be some confusion over why Armagh (a town in Northern Ireland) is in this display. To help clear up confusion, here are some useful definitions: *England*, *Ireland*, *Scotland*, *Wales*, and *N*orthern Ireland are all different *countries* and different *nations*. *Great Britain* is the geographical island, consisting basically of England, Scotland, and Wales. *UK* is short for *United Kingdom*, which is short for *United Kingdom of Great Britain and Northern Ireland*. It consists of England, Scotland, Wales, and Northern Ireland and is a sovereign state. The individual countries within the state have various degrees of self-rule. Even the International Organization for Standardization confuses Great Britain and the United Kingdom, as the ISO 3166 country code for the UK is, sadly, *GB*. *Ireland*, the geographical island, consists of the sovereign state Ireland (more formally, the *Republic of Ireland*) and the previously mentioned nation of Northern Ireland. The history of how this confusion of names came to be is quite interesting but, as you might guess, somewhat long for a footnote.

Figure 6.16 shows a small subset of the available data for rain in the UK (executive summary: It rains a fair amount over the British Isles) filtered down to a few stations and using data only from 1940 onward. This chart is constructed from two overlaid graphs:

- A base graph consisting only of the single data item "UK" with a map statistic that converts the countries named into map polygons.
- A time series chart that consists of a table of data containing the variables *location*, *latitude*, *longitude*, *date*, and *rain*. *Latitude* and *longitude* are used as faceting variables to locate the individual graphs, which are simple time series of date by rain. *Location* is used to label each time series *l*ine element

To ensure the overlaid charts match up, they have both been given the same display bounds, and the range of the x and y dimensions have been set to the same values, ensuring that the mapping from data to coordinates is the same. In this figure we have set the individual charts to share the same coordinate system (the same x and y ranges for each chart) to make it easier to compare the absolute values of the time series. We can see that the stations on the west coasts receive considerably more rain than those in the east, but we cannot see similarities between the stations; indeed there is not much evidence of strong patterns in any of the series at this scale of display.

Figure 6.17 further filters the data to just the year 2007. At this level of detail, we can see more correlation. Cardiff and Eastbourne are similar, as are Armagh, Durham, and Braemar. Tiree and Paisley form another group. A clustering analysis was performed on the data, and although it confirmed the general pattern when restricted to data for 2007, it showed a somewhat different pattern for the complete data set, as shown in Fig. 6.18. This view shows the strengths of the correlations between the time series at each weather center, and the fitted layout does not produce strong groups but instead a more smooth distribution that, on comparison with Fig. 6.17, appears to match the geographic locations quite well. The human ability to spot patterns is very strong and can lead to patterns being perceived that are not truly present. Where at all possible, it is helpful to try and show an effect (such as the apparent correlations) in a different chart so as to provide a little more evidence that things are as they seem.

The technique used in Fig. 6.18 is similar to a statistical technique known as multidimensional scaling, for which good texts are [31] and [144]. The general idea is to create a layout of points where the distances between points are approximately proportional to a matrix of defined distances between those points. When we have a matrix of measures of association between variables, the method creates a physical layout where variables placed close together are likely to be strongly associated. This is a useful method whenever a data set contains a number of different time series – it provides a simple overview of the variables that gives an interpretable, rough-and-ready clustering.

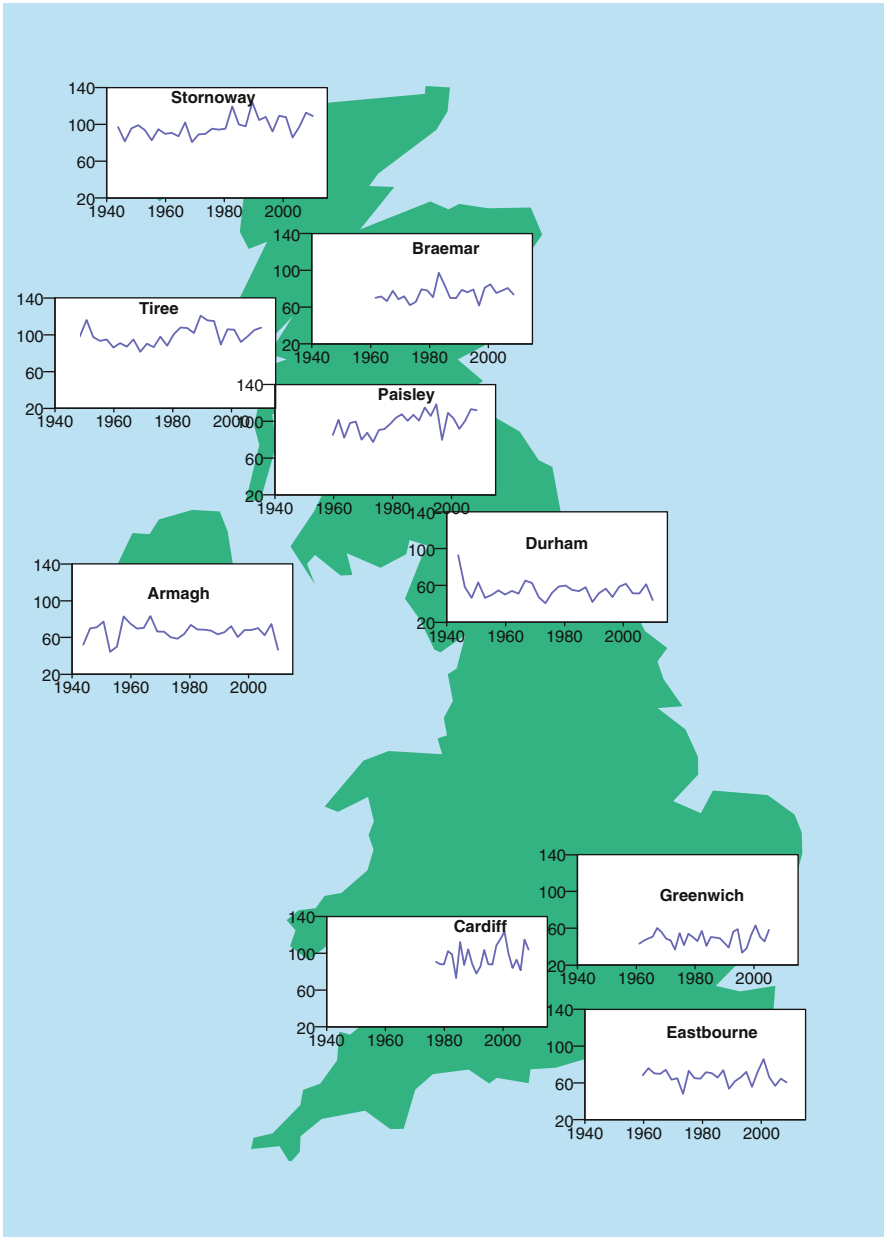


Fig. 6.16 Monthly time series showing total rain, measured in millimeters, at a set of recording stations in the UK. Although some stations recorded data back into the 1800s, only data from 1940 onward are shown in this visualization. Some stations (such as the Cardiff station shown in the plot) started recording at a later date. The time series have been faceted by location, with the facet layout being coordinate based, rather than the usual tabular or wrapped methods

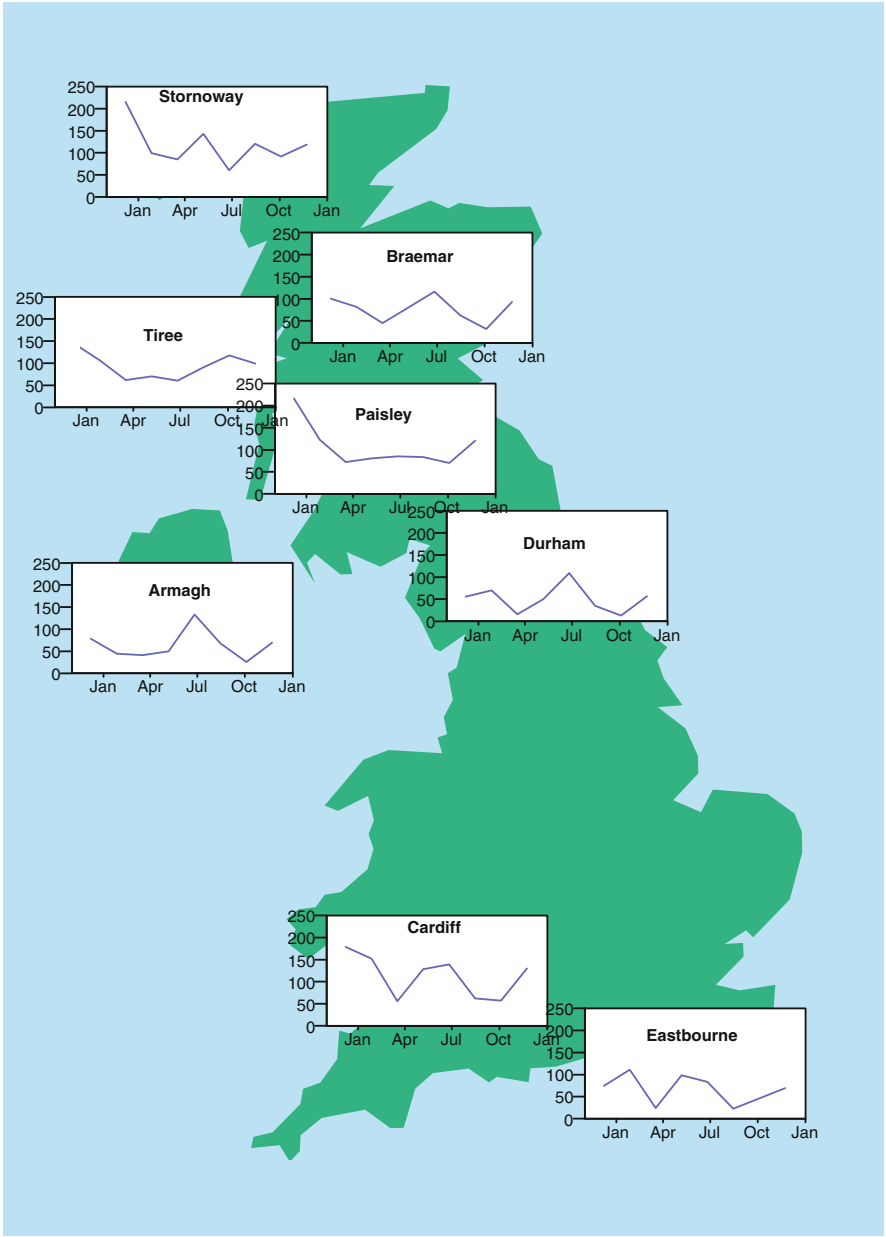


Fig. 6.17 Rainfall in the UK. This figure is similar to Fig. 6.16 on the preceding page, except that only data for the year 2007 are shown. This makes it clear that the weather patterns in the UK are dominated by the latitude of the station, with similar time series in the south between Cardiff and Eastbourne, in the middle between Armagh and Durham, and so on

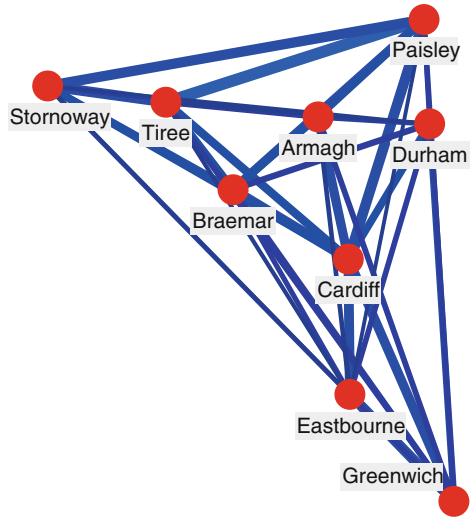


Fig. 6.18 Similarities between weather station rainfall time series. The data for this chart consist of measures of association between the time series for rainfall at each weather station. The measures are robust statistics that can detect more complex associations, but for these data they reduce to a robust form of simple correlation. The resulting matrix of correlations is laid out as a network, where the distances between variables (the stations) indicates how closely correlated they are. Since we are placing nine nodes in two dimensions (18 parameters) using data for $9 \times 8/2 = 36$ distances, the distances cannot be made to fit exactly. The color and width of the lines are used to display the actual associations

6.4.4 Faceting When Data Are Not Categorical

Time is generally considered to be a continuous variable, but faceting requires categories. We can break time into ordered categorical data in a number of ways. The simplest way is to use a measure of time that is slightly less granular than the data's natural level; for daily data we could use weeks or months; for monthly data, years; and so on. Figure 6.13 on page 140 used this technique, taking data recorded every 15 minutes and faceting by the day component of time. An alternative technique would be to bin the data, as is done for histograms, but care should be taken that the bins make some sort of sense – bins every 36 months would be fine, but bins every 35 months would be irritating.

One disadvantage of splitting a time series up in this way is that it is harder to see patterns when they occur across breaks in the data. For the passenger arrival data, this is not an issue because there is a natural split at night time, but for many series such a split could make plots harder to interpret. One solution to this problem is *shingling*. Shingling is an important part of the *trellis* display paradigm. A trellis display is a variation on structured faceting; essentially it is a faceting with a special form of axis drawn for the facet variables. Trellis displays were introduced in [5], and [109] provides a good discussion of their use, contrasting them with interactive methods for conditioning data.

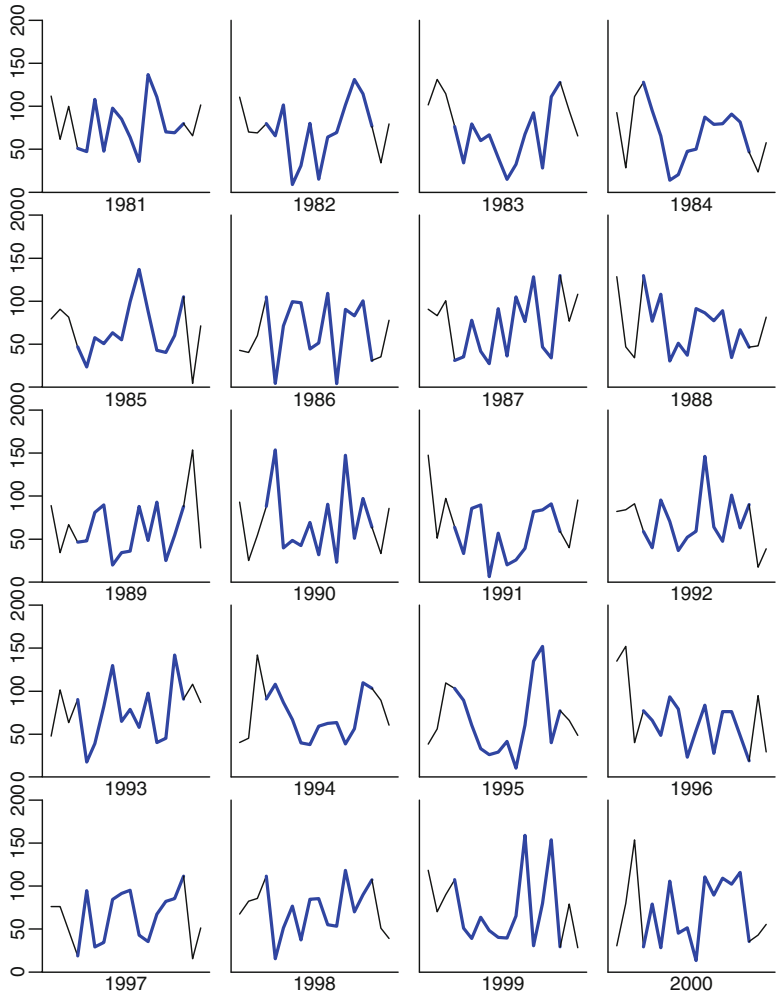


Fig. 6.19 Rainfall in Armagh. The rainfall data have been filtered for this display to show only data for the weather station in Armagh for the last two decades of the second millennium. The single time series has been paneled by the year component of the time variable and shingled with a window of 3 months. The main data for each panel are shown in *blue*, with the shingled data for the previous and next 3 months shown with a thinner *gray line*

The concept of shingling can be used both with and apart from the trellis display methodology. A continuous variable that is to be used in a faceted chart is broken up into groups as described above, but the data values at the edges of the groups are duplicated in the groups that they adjoin. Figure 6.19 shows the rainfall in Armagh, with time shingled into yearly data. As well as the data for the year defined by the panel, each panel also contains data for the previous 3 months and the following 3 months, so the time series for each "yearly" panel is actually 18 months long.

The "real" data are shown in red and the duplicated data from other time periods are shown in blue. This means that the last third of each chart is the same as the first third of the next chart (with colors reversed). Shingling is a valuable technique that allows patterns to be detected more easily, with the slight disadvantage that since some data are duplicated and other data are not, some data are given more visual importance than others. On balance, shingling is a desirable feature when there is any expectation that patterns between panels may be of interest.

6.5 Summary

Choice of coordinates forms the single most important decision made when generating a visualization. Our cognitive systems excel at judging relative distances and lengths, and so we should choose the most important variables to map to coordinates. Since you are reading this book, the assumption is that time data are of strong interest to you, and so using time as a coordinate is a good choice. Furthermore, since time is almost always an independent variable (we use it to predict other variables, rather than having other variables predict it), it is an excellent default to place it on the horizontal axis. This chapter has covered improvements and additions to the basic techniques described previously and has provided, via faceting, a way of showing time in a nontemporal chart without making changes to the basic appearance of the chart. This is done by placing time *outside* the chart. In Chap. 7 we will show how to add time to a nontemporal chart by placing time within the coordinates of the chart. These three techniques – time as a coordinate, time as faceting, and time as an aesthetic – provide the basic functionality needed to effectively visualize time.

6.6 Further Exploration

Stephen Few, in Chap. 7 of his book *Now You See It: Simple Visualization Techniques for Quantitative Analysis* [38], describes a number of techniques for time series data. He discusses animation as an analysis technique, as well as touching on several other topics related to portrayal of time data, but his comments and suggestions on scales, faceting, polar transforms, and other topics covered in this chapter stand out as excellent material. The book (a companion to [37]) is a great read in general.

One strong area of personal interest for me is spatiotemporal data visualization. It is a tricky and complex area for visualization and undergoing much active research. Articles such as [100] show a different approach to that taken in this chapter and bear strong consideration. Chen and MacEachren [19] use a more standard approach along the lines discussed here.

Chapter 7
Aesthetics

My conception of time – my ability to distinguish between consecutiveness and simultaneousness – seemed subtly disordered; so that I formed chimerical notions about living in one age and casting one's mind all over eternity for knowledge of past and future ages.

H. P. Lovecraft, *The Shadow Out Of Time (1934–1935)*

7.1 Time as a Main Aesthetic

Aesthetics were introduced in Sect. 2.4 on page 41. They provide a simple way to augment an existing chart by adding a mapping from a variable to a graphical property. Like many simple principles, there is a fair amount of complexity in the details, which are explored in what follows. To illustrate these principles we will use data on the activities of medieval English soldiers. These data has been retrieved from the UK's Arts and Humanities Research Council-funded project *The Soldier in Later Medieval England Online Database* [114], a fascinating searchable archive containing just under 90,000 records of English soldiers for the years 1369–1453.

The base data consist of per-soldier records that span multiple years and give information on what the soldier did and who he served with. An example record records data on a soldier named `John Willis`, whose status was a `Yeoman/Valettus`. He served as an `archer` under `Thomas Neville, Lord Furnival` in the period `1404--1406`, during which time he was part of a `standing force in north Wales`. This level of detail might be fascinating for a genealogical lookup, but to answer quantitative questions, the data need to be manipulated. First, a separate record for each man/year was created, so that each record stood for the same basic quantity: the employment of one soldier in one year. The data on status was simplified, dropping additional terms so that, for example, `Yeoman/Valettus` became simply `Yeoman`. The `NatureofActivity` variable was used to generate two fields – a general `Activity` field that defined the *type* of

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_7,

Table 7.1 Data on medieval soldiers who served under Henry V during his 1415 expedition into France, at which he fought battles at Harfleur, Calais, and – most famously – Agincourt

Year	Country	Activity	Rank	Status	Commander	Count
1415	France	Expedition	Man-at-arms	Baron	Henry V	1
1415	France	Expedition	Man-at-arms	Duke	Henry V	3
1415	France	Expedition	Man-at-arms	Earl	Henry V	11
1415	France	Expedition	Man-at-arms	Knight	Henry V	54
1415	France	Expedition	Man-at-arms	Esquire	Henry V	313
1415	France	Expedition	Man-at-arms	Unknown	Henry V	1097
1415	France	Expedition	Unknown	Esquire	Henry V	7
1415	France	Expedition	Unknown	Yeoman	Henry V	15
1415	France	Expedition	Gunner	Unknown	Henry V	2
1415	France	Expedition	Archer foot	Unknown	Henry V	832
1415	France	Expedition	Archer	Yeoman	Henry V	1355
1415	France	Expedition	Archer	Unknown	Henry V	3687

activity and a `Country` field that defined *where* it took place. Then the records were aggregated together to create a table of all unique combinations, together with their counts. This table consists of 602 records – Table 7.1 shows a sample of these records associated with King Henry V's expedition into France in the latter half of 1415.

As with any data on employment and manpower, good questions to ask are:

- What are people doing?
- Where are they doing it?
- When do they do it?
- Who is doing it? How many people are doing it?

Figure 7.1 has been created to answer those questions. "What?" and "Where?" are the most important questions, so they have been used in the chart to encode the glyph positions. The time factor is less important, so rather than add it as a coordinate or as a faceting variable, we instead use color to encode the year. This gives us the basic plot. However, that plot would be misleading. Each point in the plot represents a group of soldiers – and those groups are of very different sizes (as Table 7.1 shows). If we do not take into account the count information for each group, we will give equal visual prominence to a single baron as to 1355 yeomen. Perhaps for Henry V this would be reasonable, but we are more democratic and want a display that gives equal weight to each person. To keep the graph truthful, we must therefore add an aesthetic that compensates for the group sizes, so we use a size aesthetic to indicate the number of soldiers in each group.

7.1.1 Representing Counts

As a slight aside, using size for counts is a good standard technique. It is easy to remember, and using size for counts means that the visual "weight" of a symbol

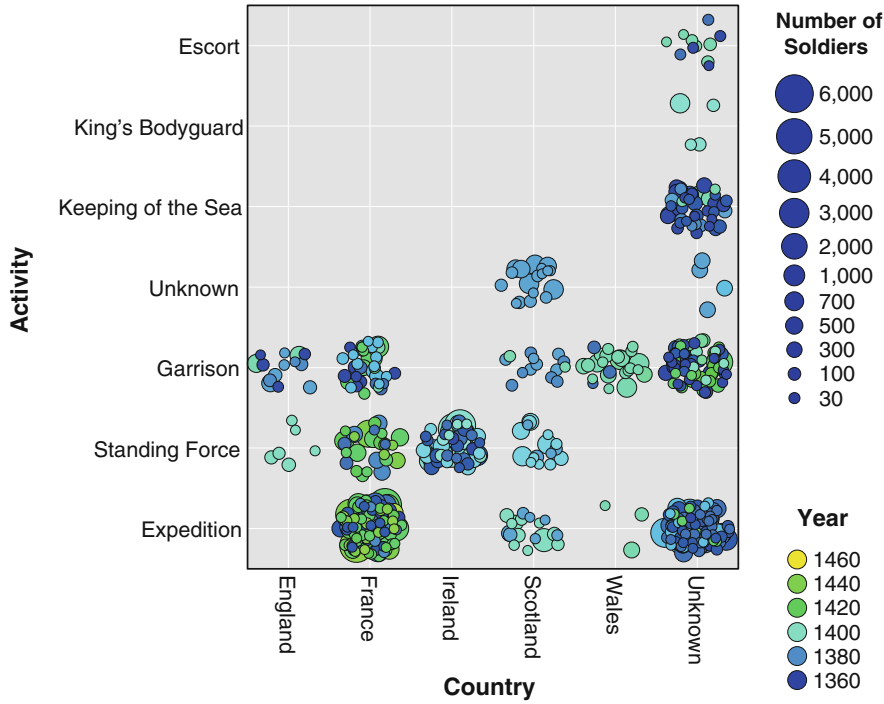


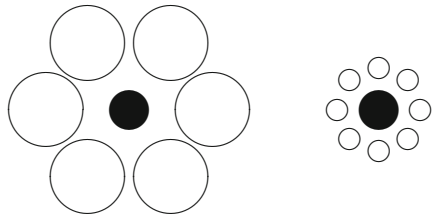
Fig. 7.1 This chart shows which activities were performed by English soldiers and in which countries those activities took place. The data are shown as a scatterplot of activity by country, so each record denotes many soldiers. The number of soldiers is shown with a size aesthetic, and the color maps to the year of record. Jittering has been applied to the *points* so they occlude each other less

in the graph[1] corresponds to what a statistician would be likely to use as a weight variable when defining statistics or models for the data. One point to keep in mind is that it is helpful if count is mapped to area, so a root transform is often needed if a chart defines size using a radius. Several amusing examples of blatantly ignoring this rule are given in [57].

Even with reasonable care taken, human perception is not well tuned to the task of matching sizes to counts and can be fooled easily. Studies indicate that people overestimate small circles' areas and underestimate large circles' areas. Another well-known example is that of the Ebbinghaus illusion – the perceived sizes of a

[1]One of Tufte's [111] criteria for evaluating charts is the *data–ink ratio* – the proportion of ink corresponding to the data as opposed to other elements. This is a different criterion. What is urged here is that if a measure of size is being represented using an aesthetic, then there should be a correlation between how noticeable the element is and the value of the size variable. Bigger, more contrasting elements are more noticeable, and so we say they have more *visual weight* than smaller, less contrasting elements.

circle depends on sizes of the surrounding circles, as was shown by Titchener [110] in 1901. In the graphic below, both filled circles are the same size, but they appear different.



Overall, using size to represent counts is a good idea – for simple extents, such as shown in a bar chart, it is very effective. When area is being portrayed, judgement can become harder, but it is still a strong technique. In three dimensions, it gets increasingly hard to make good comparisons – avoid volume comparisons if you can.[2]

7.1.2 Summarizing and Splitting Aesthetics

Section 6.4 on page 136 introduced a systematic way to define a chart; start by identifying the variable of interest, then use coordinates, faceting, and aesthetics to split the data apart into groups and allow comparison. In Fig. 7.1 the activity has been split by country to give the basic plot, and then we split each of those groups into many smaller groups, one for each year.

This shows the use of an aesthetic as a *splitting aesthetic* – an aesthetic that defines a different group for each value it takes on. The alternative is a *summarizing aesthetic* – an aesthetic that shows, for a single glyph, a summary of the data it contains. If we rework Fig. 7.1 to use time as a summarizing aesthetic, we create Fig. 7.2. For this visualization, we have also added a third aesthetic, aspect, which summarizes time using a different summary. The aesthetics in use are as follows:

- Size – the area of an ellipse shows the total number of soldiers represented by the glyph. It is a summarizing aesthetic using the sum of the counts as the summary statistic.
- Color – the color of an ellipse shows the average year in which the soldiers represented by the glyph were active. It is a summarizing aesthetic using the mean of the year as the summary statistic.

[2]It is generally a good idea to avoid 3-D anyway. Since we are not good at judging position in 3-D, and our perception of 3-D is highly conflated with how we perceive aesthetics such as size, brightness, and saturation, it is best only to use it for very simple encodings, or just for decoration.

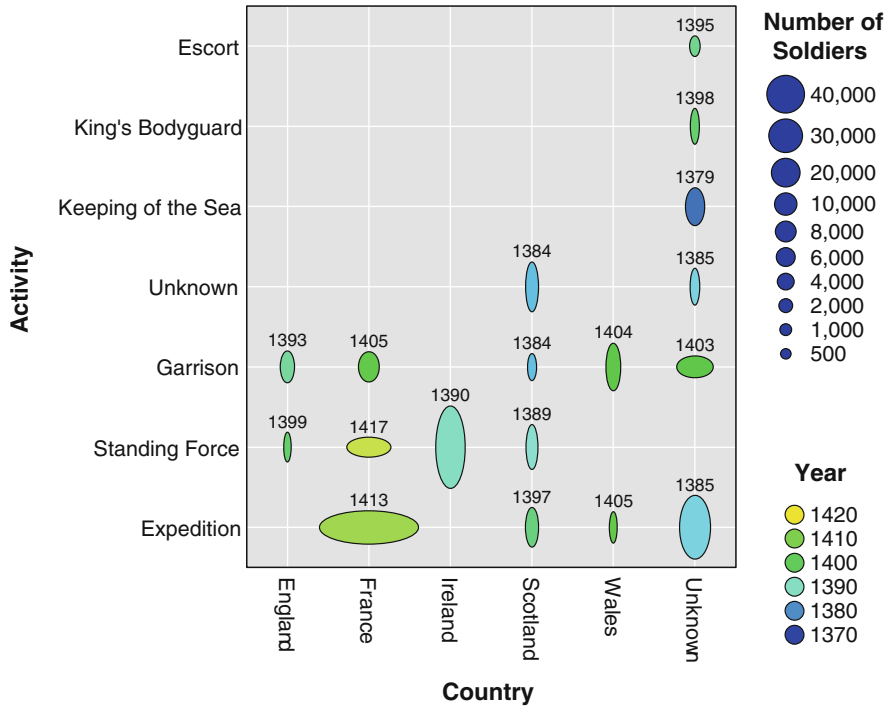


Fig. 7.2 This chart shows which activities were performed by English soldiers and in which countries those activities took place. The data are shown as a categorical scatterplot of activity by country, aggregating the data so that one *glyph* only is shown for each unique {activity, country} pair. The number of soldiers has been shown with a size aesthetic, color, and label map to the average year of record, and the aspect ratio of the ellipse indicates the standard deviation of the year variable for records in that aggregation, with *narrow glyphs* indicating small deviation (a narrow time span) and *wide glyphs* representing large deviations (wide time span)

- Aspect – the aspect ratio of an ellipse shows how spread out or narrow was the time span in which the soldiers represented by the glyph were active. It is a summarizing aesthetic using the standard deviation of the year as the summary statistic.

Looking at the figure, we can see that France is different from the others – activities there tend to be more spread out than for other countries. We can see that "Keeping of the sea" was an early activity and that activities in Scotland and Wales had a very short focus.

This chart has a potential weak point – two aesthetics that are related (size and aspect) are being used for different variables (count and year), and year is being represented by two different aesthetics (aspect and color). It is generally better to keep the separation cleaner and use color to represent the counts, leaving size to give

mean(year) and aspect to give *standard deviation(year)*. However, as Sect. 7.1.1 indicates, using counts for sizes is so natural that it trumps the desire to keep a clearer match-up between variables and aesthetics.[3]

The importance of understanding the difference between summarizing and splitting data is best seen in the context of data size. Historically, only small amounts of data were collected, and plots that displayed a single glyph for each record were acceptable. In that tradition, all the data were automatically split into individual rows and so the difference between a summarizing and splitting aesthetic was irrelevant – if you have one record, it neither gets split nor needs summarizing. In contrast, database techniques invariably assume that you want to "roll up" data into groups and summarize the groups. A brief look at any SQL book ([41] is a good introduction) indicates this, as does the prevalence of database systems such as OLAP (OnLine Analytical Processing) that *only* work with aggregated data. A SQL specialist looking at Fig. 7.2 would immediately think of a statement similar to

```
SELECT MEAN(year), STDDEV(year), SUM(count)
 GROUP BY activity, country
```

If you have a large data set, or if you are using a database (or, more likely, both), then a good principle is to start with summarizing aesthetics and to move to splitting aesthetics only when it is possible to do so without generating too many groups.

7.2 Specific Aesthetics

The previous section showed examples of the use of color and aspect ratio to represent time and statistics on time. Time, as we showed in Chap. 4 is an interval variable – distances between time points have meaning, but there is no natural zero point and time points cannot be added. Elapsed times, on the other hand, are ratio level and can be used more freely. We made use of this in Fig. 7.2 by using the mean value of times (mean can be applied to an interval variable) and by using standard deviations of time (which results in an elapsed time) for the aspect ratios. Another point brought out was the importance of considering related aesthetics. The following sections will elaborate these points in the context of different classes of aesthetic.

7.2.1 Coloring by Time

Since the 1980s color has been readily available for graphical displays. Color has the useful property of keeping the physical layout of the chart intact while

[3]The process of designing a visualization often involves compromises such as these. The guidelines of Chap. 3 give suggestions on how to decide what is important and how to prioritize, but in the end judgement calls must be made.

not interacting with the mapping from variables to coordinates. "Color" is also a complex aesthetic, both technically and perceptually. We use the term color to mean hue throughout this book, although the latter is more technically correct, as the general term color includes the aesthetics hue, saturation, brightness or lightness, and transparency. There are many resources available on the appropriate use of color in graphics (e.g., [11,12,72]), and the subject was touched on in Sect. 2.4 on page 41, so in this section we will concentrate on time-specific applications.

Hue is a seductively easy aesthetic to use and works well for time. One reason is that time data do not have an origin, and neither does hue. It forms a circle, which is often the way we want to think of time. The heat metaphor can be used to good effect for representing time by hue; as time goes by things cool down, and so older items can be represented by cooler colors. New items are called "hot," and so it makes sense to use a color scale that matches this mapping. In Figs. 7.1 and 7.2 we have used a hue scale for year that has the oldest times in blue and the most recent ones in yellow.

Saturation, lightness, and brightness are not as good for basic time data. They all have a strong, well-defined zero point and so are more suited to ratio-level measures. They can be used for elapsed times (also called time durations), as a duration of zero is well defined and meaningful. We could have substituted any of these aesthetics for aspect in Fig. 7.1, for example.

Transparency is generally a poor choice for time. Although a color aesthetic, it has much more of a "size" feel to it – using it tends to make the element appear weighted, and weighting by time is a useless concept. Weighting by duration, on the other hand, makes more sense, so again, it can be used for time durations to good effect.

In many figures in this book a mixture of the above are used in the same display, with hue and saturation being changed simultaneously. One particularly useful way to use such a combination color aesthetic is with a *double-ended scale*. For such a scale there is a clearly defined zero point that divides the scale into two sections above and below this point. In time situations, we might have *now* as the zero point, with *past* and *future* being the two periods. A neutral color (black, white, or a shade of gray) is used to show that zero point, and two distinct hues, or families of hues, are used to show the two parts of the scale. The intensity of the color mapping increases for values further away from the zero point. Maps often take advantage of this technique, using color for the height above or below sea level, but it works well for any situation where there is a midpoint for a variable that is important to the viewer. Examples of double-ended scales can be found in [13, 14], where they are termed *divergent* scales.

7.2.2 Sizing by Time

Size-based aesthetics, including width, height, depth (in three dimensions), and aspect ratio, are best reserved for measures that are naturally associated with size,

such as sums, counts, and weights. Time is not a good match for this, and so it is usually best to reserve size aesthetics for time durations – and even then only when it makes sense to use those durations as a visible weighting.

The use of size for time is occasionally useful when no other aesthetic can be used. Size comparisons, especially 1-D comparisons such as width or height, can be performed quite well by most people, and so if you need an aesthetic for time, it is at least relatively easy to use, if not a naturally appealing mapping. An example of this is given in Fig. 7.3, in which the year is shown using a width aesthetic. We show later time points as larger, which is a more natural ordering based on informal discussions; as a product of our evolution, we tend to think in terms of natural order, and in the natural world, things grow over time, rather than shrinking.

Figure 7.3 focuses on who, where, and how many. We see commanders like Scrope and Stanley, sitting in Ireland with the same sized forces for three or four sequential years, and can compare them to Henry V, Gloucester, and York, who have a single powerful year in France as commander.[4] The year of command is a secondary feature – we can cross-reference to see that the Irish commands were in the 1300s and the French ones in the early 1400s, but we cannot see the 2-year difference between Henry's and Gloucester's commands.

Figure 7.4 shows a variation on the previous figure. We drill into France (filter all non-French commands) and then have freedom to use another variable for color. We pick the soldier's rank, and the resulting visualization allows us to see the difference in composition of armies. Here, the relationship between time and the other aesthetics is more interesting. A clear difference can be seen in the soldier's ranks as time progresses. The thinnest bars (oldest armies) have a fairly even mix of archers and men-at-arms, but by the time of the battle of Agincourt in 1415, Henry's army was mostly archers, a situation which continued under Gloucester in 1421 (and where it looks like Henry's archer foot was reclassified as simply archer).[5]

Using width for time is close to a last resort – here it is done because nothing else suits, not because it the best solution. This technique should be used rarely, and with care.

7.2.3 Shaping by Time

Shape aesthetics include aesthetics that map data to symbols or glyphs, to images and to line dashing. These are aesthetics that are categorical in nature. They produce

[4]Henry V had campaigns in 1415, 1417, and 1421 in France, but the records give Gloucester as the commander in 1417 and do not record the commanders in 1421. These campaigns were all under Henry as king, but only the 1415 expedition records him as being the commander.

[5]Another reason for the increased number of archers might be that archers were paid considerably less than melee combatants, due to their lower risk of death. As any good manpower study would have told Henry, if you can get the same amount of effective work from cheaper workers, it makes sense to use them as much as possible.

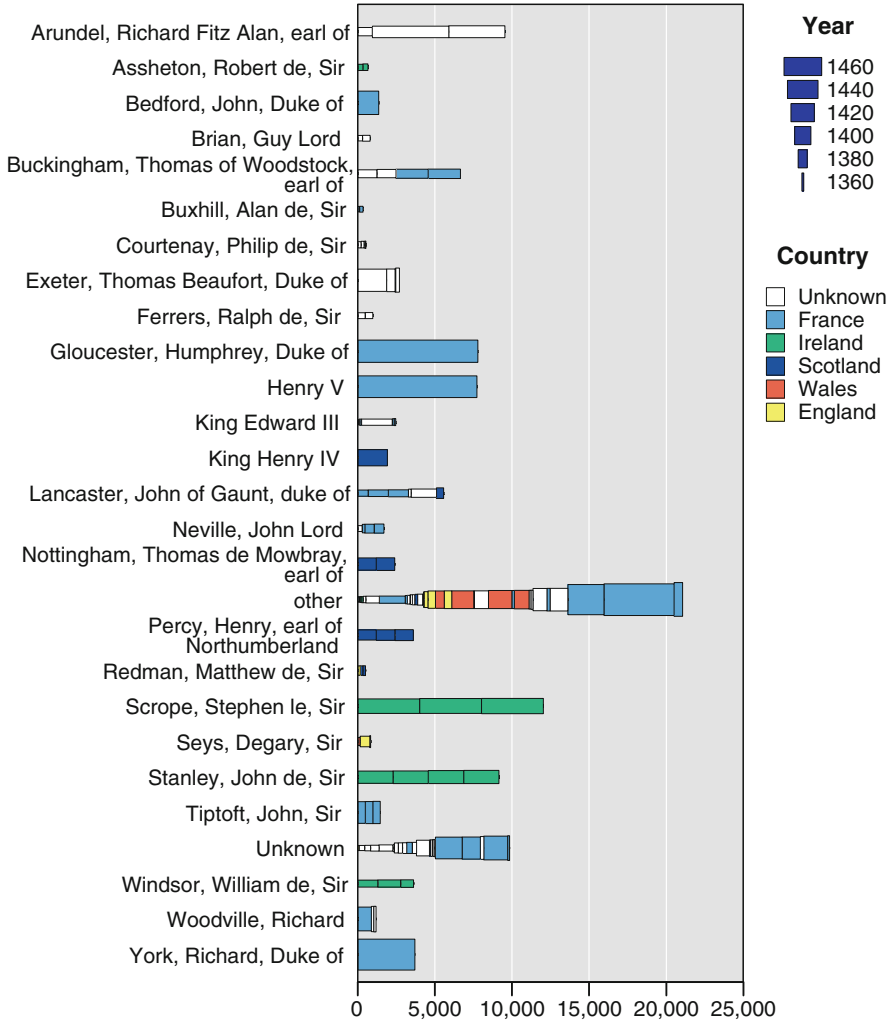


Fig. 7.3 Medieval soldier: commanders. This figure shows the total number of men under each leader's command. The bar chart of commander by count has had each *bar* split into groups both by year and country. The country in which the command was held has been mapped to color and the year in which command was held to width. Thus Scrope, for example, led a force of about 4000 men for three different years, always in Ireland. It is clear that commanders do not lead in different countries often, with John of Gaunt being the main counterexample. He campaigns in both Scotland and France. "Unknown" refers to records where no commander was given, and "Other" commanders is a catch-all category formed by aggregating all commanders who commanded only a small number of men in their careers

distinctly different items, and so to use these for a general time variable we must define a transformation from continuous time to categorical time, as we show in Sect. 8.3 of Chap. 8. It is possible to work with continuous data – we could specify

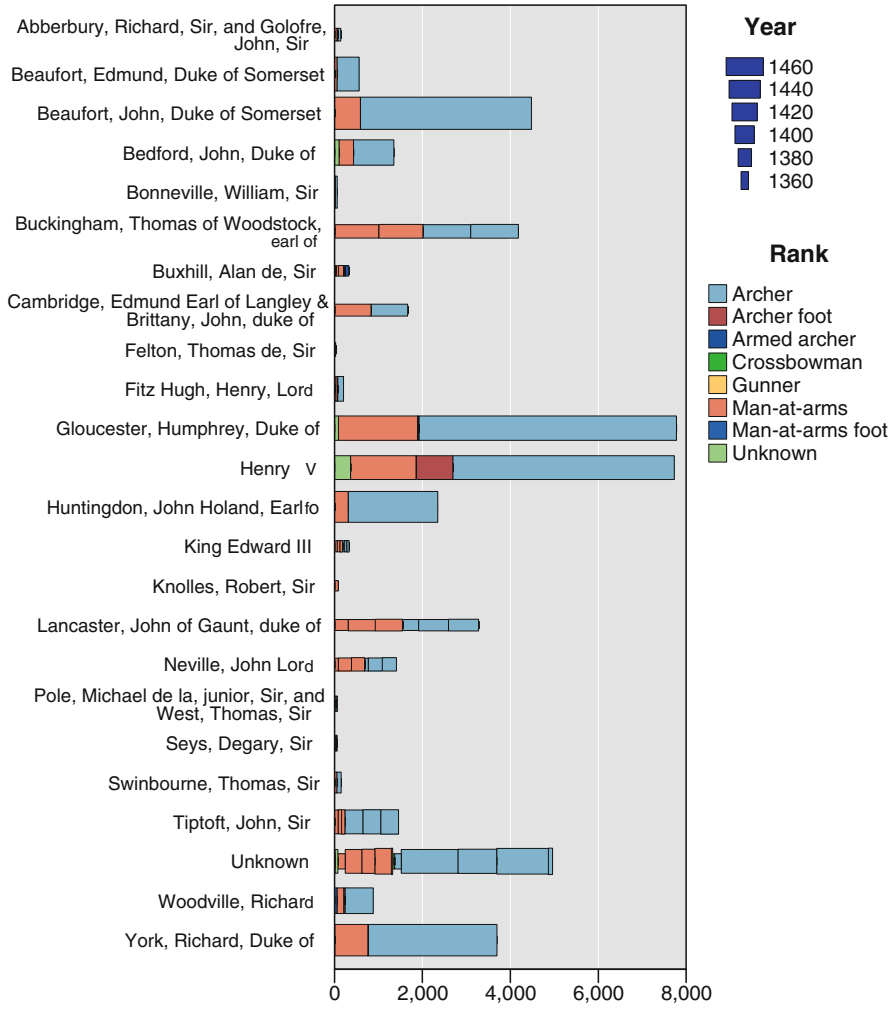


Fig. 7.4 Medieval soldier: French commanders. Similar to Fig. 7.3 on the previous page, this figure shows a stacked bar chart of commanders, but it has been filtered to those commands that took place in France. Year is mapped to width, as before, but color is now used for the soldier's rank, allowing us to see that, although early commanders (John of Gaunt; the joint command under Cambridge and Brittany; Thomas, Earl of Buckingham) had a near 50–50 split between archers and men-at-arms, later commanders had a much higher proportion of archers

a dash pattern where dash granularity depends smoothly on a variable, for example, but our perceptual system will not support this technique, and it will not be effective.

An important case to consider is time data that are already categorical in nature. This occurs when we have time data that represent slices of time, such as *day of week* or *month of year*. These can be treated as categories, although since they are ordered (often cyclically), it is a good idea to find mappings to categories that exhibit that

ordering. In fact, this same advice holds true for other aesthetics, and so for data of this type, the cyclical rainbow hue scale, which is generally not advised as a perceptually useful tool, works out fairly well.

7.2.4 Other Aesthetics and Time

Many other aesthetics are possible. In this book we show label and visibility aesthetics, pattern and glyph rotation aesthetics, but many others could have been shown. The previous sections' advice will hold for any other aesthetic. The major points to consider are whether the aesthetic is categorical in nature, whether it tends to be perceived as a visual weight, and how the aesthetic output can be mapped to show time's strong ordered nature.

Another useful technique is to have aesthetics on aesthetics. If labeling is considered an aesthetic, then a number of techniques work well that modify that aesthetic:

- Changing the font size of a label in proportion to a measure of size for the variable;
- Adding italic, bold, or color for "flag" variables – variables that indicate the presence of a feature;
- Using a visibility aesthetic on labels only to label certain elements.

In the following section, we use aesthetics on text labels extensively.

7.3 Time as a Secondary Aesthetic

In this section we will use as motivation the following data set: The original data consist of the text contents of a document (in this example, a book). The book is considered as a sequence of words, with uninteresting words (the *stop list* or *stop words*) removed from the sequence. This sequence is then further reduced so that only a small number of the most commonly occurring words are retained. The goal of the visualization is to give an overview of *what* the book is about and *how* it is structured. Clearly, the words together with their frequencies are the most important information, and so a natural suggestion would be to make a bar chart, which allocates position to both variables. However, this will not work, as the data set has 300 words of interest, and to display that many words we need another technique.

We could just list the words, sorted by their frequency order (with the most frequent first). That would give us a simple list like the following one, using Rudyard Kipling's *The Jungle Book* [67] as the document.

```
mowgli man jungle head bagheera back big nag rikki wolf tikki
time men baloo sea khan night people shere toomai elephants
kotick good long till pack eyes feet great knew mother cub
```

```
made heard akela kala elephant told father kaa things day
kill seals white young brother looked run water began half
wolves round seal teddy horse place wild fight ground monkeys
tail end black boy called give hunting make mule nagaina put
red dead thing tiger afraid full gray neck sahib sat troop
billy law rock side tails village catch eat killed left log
ran snake tree bandar council darzee foot gun petersen art
children cubs found camp hunt song talk buffaloes dance gave
herd life remember ten angry bullocks deep fighting grass
hundred son speak call skin teeth branches dark death fire
hear island miles stood trees work year cattle caught house
shouted tabaqui thought camel find grew lay turn years heavy
met mouth ravine true turned word words beach buldeo bull
cried hard hills panther sleep wise bad beaches cave child
ears fro heart lines mongoose monkey shoulder dog eye free
guns hill minute moon show twenty wall beast close dropped
felt hands hide holluschickie killing noise novastoshnah
room coming cow earth friends garden hand king legs quiet
shoulders sit broken brown care cold dry face follow foolish
held lie line matkah matter means order play set slept voice
wait air bear break brought bulls clear eggs front hast hind
hold leave live lying morning mud nose ready running sides
stopped taught warm world battery born bring brothers coat
days driver fear forest forward frog green hot hurt jump
leg listen lost making master mind move path plain shut stop
swim war wife answer bed branch breath broke carried charge
city climb cobra cut drive fast forty
```

These ranked frequencies are of some use, but not knowing their frequencies is annoying. We can restore the ability to make a frequency comparison by using a size aesthetic mapping the frequency to the total area used to represent the word.[6] However, this leads to a display that contains a lot of unused space, and the representation is not compact since there will be a lot of white space around small words that are on the same line as bigger words.

To solve this we give up on the ordering of the words, which is simply a redundant encoding if we are using size for the frequency, and use an algorithm that places the words within a rectangle to make a "cloud" of words, where the position of the words has no meaning and has been defined simply to minimize the space needed to display all the words (or, equivalently, to maximize the font size and make the display as readable as possible given a fixed size). This display, shown in Fig. 7.5, is known as a *tag cloud*.

At this point, the basic display is complete. The graph shows the main content of the book, and the result is legible and intelligible. But we do have one more piece of information that is of interest – the locations of the words within the book. For each word, we have a list of the occurrences. We could summarize that in a number

[6]We could also, more simply, use the font size to show the frequencies, but this would mean that long words like *holluschickie* would gain far more visual prominence than short words such as *hide*, whereas for these data their frequencies are almost identical, and so they should appear visually similar. This turns out to be a tricky perceptual issue; our perception of the sizes of words is not easy to quantify.

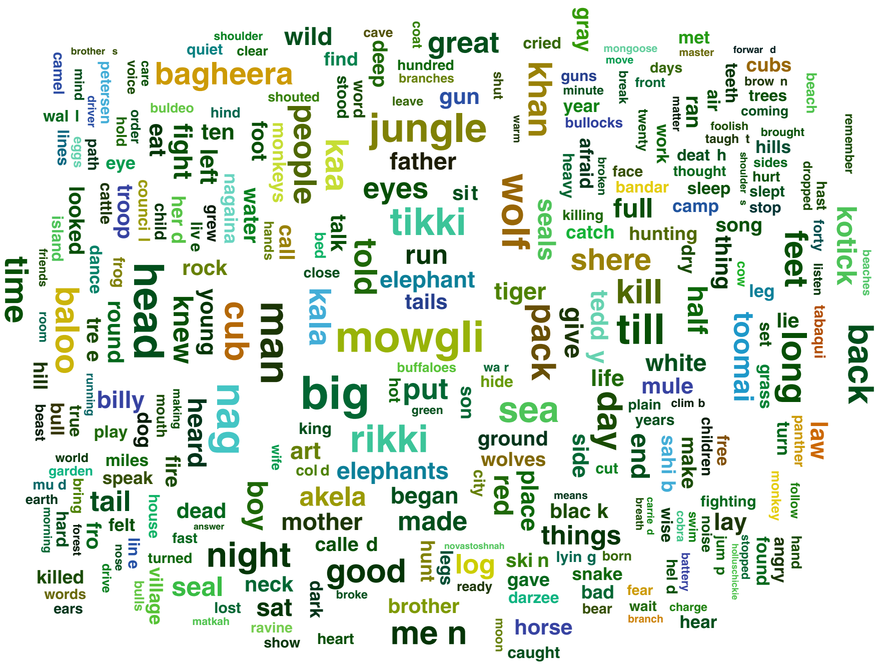


Fig. 7.5 Tag cloud showing the 300 most frequent interesting words in Rudyard Kipling's *The Jungle Book* [67]. The area spanned by each word is proportional to its frequency. The color indicates the average location within the book, with *red words* occurring near the start, *green* in the middle, and *blue* toward the end. The brightness of each word is inversely proportional to the standard deviation of the locations of the words within the book. Thus *brightly colored words* occur in a localized context and *darker words* are spread throughout the text

of ways – in Fig. 7.5 we use the mean location and a measure of the spread of the locations to give an idea of where each word is used in *The Jungle Book*.

Using color hue for a continuously valued variable is often a poor choice, as our perception of color is quite erratic and hard to design for, but in this case we are not looking for accurate quantitative results – our goal is not to be able to compare mean locations. All we want is some idea of the rough location within the book, and to know when words occur relatively close to each other. In Fig. 7.5, the color aesthetic makes groups of words stand out, allowing us to to see words grouped by color into sections of the book:

- The words `law`, `tabaqui`, `free`, `cub`, and `cubs` occur at the beginning of the book, as the main narrative, concerning the boy Mowgli and how he became a member of a wolf-pack, commences.
- The words `bagheera`, `baloo`, `panther`, `tiger`, `akela`, `shere`, `khan bandar`, `log`, `monkeys`, and `kaa` appear soon after, as more characters are introduced. The characters Bagheera and Baloo are linking characters; their colors are between the reds of the initial cycles and the yellows of the following stories.

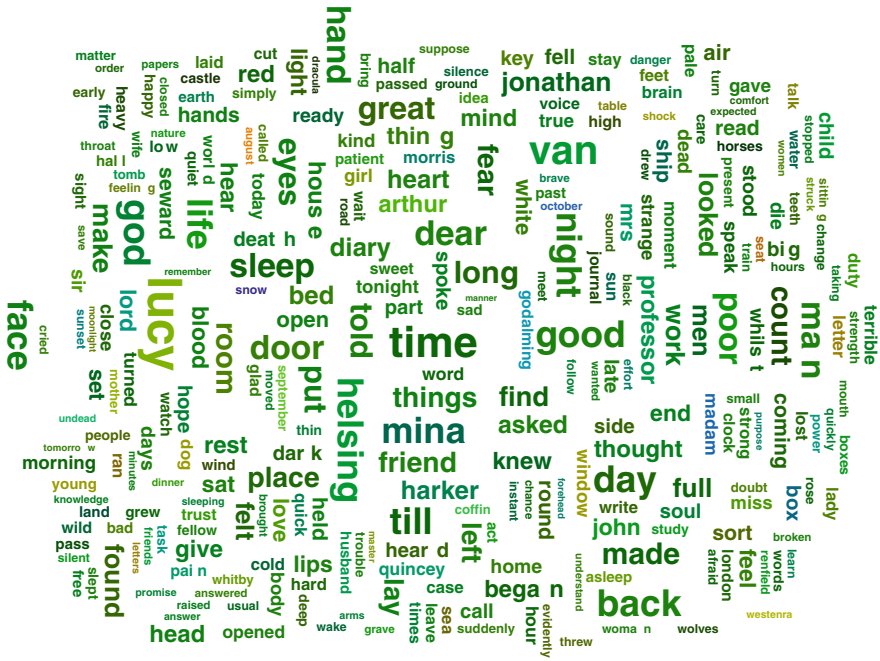


Fig. 7.6 Tag cloud showing the 300 most frequent interesting words in Bram Stoker's *Dracula*. The area spanned by each word is proportional to its frequency. The color indicates the average location within the book, with *red words* occurring near the start, *green* in the middle, and *blue* toward the end. *Dark colors* are words that are spread throughout the book; brighter words are localized to specific places within the text

- The next set of stories, in green, involve `rikki`, `tikki`, `kotick`, `seals`, `sea`, `beach`, `herd`, and so on.
- Then we see a section (in cyan) featuring the terms `toomai`, `sahib`, `elephant`, `elephants`, `peterson`, `kala`, `nag`, and `forty`. These are from a story "Toomain of the Elephants" that is not part of the more well-known *Jungle Book* stories featuring Mowgli and made into a film by Disney.
- Similarly, a set of words in deeper blue (`horse`, `mule`, `billy`, `troop`, `driver`, `camel`, `battery`, `gun`) are from the story "Her Majesty's Servants," which finishes *The Jungle Book*.
- Darker colored words are ones that are spread throughout the book. They are terms common to many of the stories and include `man`, `time`, `night`, `day`, `good`, `heard`, `eat`, `afraid`, etc.

The Jungle Book is a collection of stories, which leads to important words being more localized than in other works. Compare Fig. 7.5 with Fig. 7.6, which has been constructed identically using frequency data for a different text, Bram Stoker's novel of gothic horror, *Dracula* [106].

In this figure, the words are not as localized as in *The Jungle Book*. That is clear from the much darker colors and lack of many bright terms. The character names, which in *The Jungle Book* tended to be located in distinct areas, are here shown in darker colors close to the central green – they occur throughout the book. Lucy Westenra is the earliest casualty – she appears early in the book, but not later, due to the necessities of the plot, and the minor characters Quincey and Godalming (one of whom apparently uses the term *madam* quite often) appear later (possibly in October). A few other terms are localized, but by and large, most of the important terms in the book are spread throughout it.

One interesting feature of the novel is how important time is! *Time* itself is the most frequent word in the book, and of the top 300 words, 26 of them concern time:

```
time night day moment morning tonight watch late september
days today present clock passed times hour instant sunset
tomorrow october minutes quickly early past hours august
```

Understanding and visualizing time is clearly not just important to people who are working with data; it is needed by those whose tools are a hammer and stake, too!

This visualization is not a traditional one. It flies in the face of one of our most strongly stated tenets, the importance of position, by using position only to make a compact representation. It gives up the most powerful tool available for constructing plots simply to save space!

The reason that this chart works is that the situation is extreme. It is important that we show as many words as we can in as small a space as we can, with their relative frequencies the most important additional feature. This argues we simply pack the sized words in as best we can. Only when that is done can additional details be added. In this case we have augmented the chart to show the time/sequence information as a secondary consideration, and we have done so showing both the average value and the spread of the number of times the words occur.

7.4 Summary

Although our ability to perceive patterns in data coded as location is better than our ability to perceive patterns in data coded as aesthetics, aesthetics still have an important place in visualization. Color reproduction has become affordable and is used in many graphics. While the main message of a chart is best encoded using coordinates, aesthetics can be used as a secondary indicator. In particular, we can use an aesthetic as an informal filtering technique, so that questions like “are there different patterns in the relationship between country and activity depending on the range of years campaigns?” translate to noticing differences in patterns of position for different types of shape (Fig. 7.1).

Time, being an interval-scaled value, naturally is suited to an aesthetic that works well for smoothly varying values. However, our perception of the mapping from a

continuous data variable to an aesthetic is not an accurate one; we can judge order pretty well (although color perception is a tricky business), but comparisons as to the relative sizes of values based on their aesthetics is hard. The good news is that time, although interval in nature, is easily understood as an ordinal. We naturally aggregate data measured in seconds into hours or days, and so when we see an aesthetic based on time, our tendency to see an ordered set of groups of time is not a strong hindrance.

The difference between aesthetics that *split* and ones that *summarize* is important when adding an aesthetic to any chart that aggregates data. If a number of data cases with a differing value of the aesthetic variable are represented by a single glyph, then there are two possible choices:

- Split the glyph into multiple glyphs, one for each value of the aesthetic variable.
- Instead of using the raw value of the variable for the aesthetic, use a summary, such as a mean for the aesthetic.

The choice of when to split and when to summarize is not a simple one. Some factors that should be taken into consideration include the following:

- Ordinal and nominal variables are more naturally suited to splitting; interval and ratio variables are more suited to summarization.
- When the variable in question has many values, splitting may produce a complex chart with many overplotted items.
- Summarization may hide unusual values (if a measure of centrality such as the mean or median is used), or it may be dominated by unusual values (if a measure of spread such as the range is used).
- Splitting using interval and ratio variables generally requires some form of binning, which introduces a choice as to the bin sizes that must be decided.

Time data are very amenable for use by both splitting and summarizing aesthetics. Average times and time ranges (durations) are interpretable; summaries such as spreads based on standard deviations are sensible for times, although not generally for durations (because the distribution of durations is likely to be asymmetric – durations cannot be negative). For splitting, if the time variable has many different times, aggregation into larger time units is very natural – in fact it may be the most natural binning for any form of variable. If time cannot be used for a positional dimension, or if the goal is to use time to augment an existing chart, then there are many good options for using it as an aesthetic.

7.5 Further Exploration

There is a sizable literature on visual perception, the mapping from data to aesthetics and how we process the results, and suitable mappings for data. Some of these were suggested as further reading in Chap. 3.8. Bertin's *Semiology of Graphics* [8, 9] discusses mappings from data to graphical representation in detail, and any theories

about how visualizations work owe a debt to it, either directly or through the works they cite. Cleveland's *Elements of Graphing Data* [21] contains advice for low-level mappings and has been influential for statistical graphics designers.

Color is a very commonly used aesthetic. It is also a very commonly misused aesthetic. Cynthia Brewer's Web site *ColorBrewer* [13,14] was created to give good color scales for map applications, but the scales provided can be used for more than just geographic data.

To understand how the human visual system processes the world, Colin Ware's *Visual Thinking for Design* [124] is a highly readable work with direct application for visualization. All his books are worth reading; this one is a masterpiece of clarity and beauty. It describes the physical basis for our visual systems and applies it to the design of visualizations.

Chapter 8
Transformations

How amazing time is, and how amazing we are. Time has been transformed, and we have changed; it has advanced and set us in motion; it has unveiled its face, inspiring us with bewilderment and exhilaration. Yesterday we complained of time and feared it, but today we love and embrace it. Indeed, we have begun to perceive its purposes and characteristics, and to comprehend its secrets and enigmas.

— Kahlil Gibran, *Children of Gods, Scions of Apes* (1883–1931)

8.1 Distortions of Time

In most of the charts shown in this book, time is displayed in a linear fashion; each centimeter on the page represents the same extent of time. This allows us to make comparisons of durations easily, which is often of prime importance. However, there are times when breaking this rule makes sense – when we want to distort time and show it with a nonlinear mapping. Typically we will want to distort time when one or more of the following statements hold true:

- Time extents (durations) are not displayed or of strong interest – the data are more likely to be point event data.
- The length of the times between events are not of strong interest – the order or sequence of events is the more interesting aspect of time.
- Events are not distributed evenly over time, and this uneven distribution causes linear mappings of time to fail to show the data clearly.
- Because distortion mappings concentrate attention (see, for example, Sect. 9.2.3 on page 188), there is a natural time location on which to focus attention.

Figure 8.1 shows an example visualization that is a good candidate for modification. Movie releases are point events – effectively, the duration of a movie is from the point of release forward to the present day, making the extent information of no

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_8,

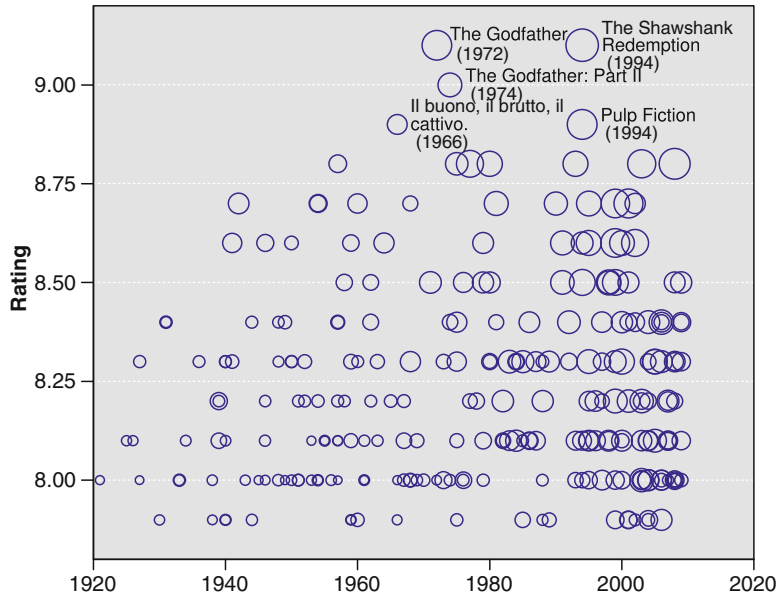


Fig. 8.1 Top 250 movies as of January 2010. This chart shows the highest rated movies as a scatterplot of ratings by time. The number of votes for each movie has been mapped to the size of the points. A square root transform has been applied to the size aesthetic so that the area represents the counts

interest,[1] and so on our movie time line, the movies are represented as point events on their release dates.

Although we could be interested in the relative densities of the top 250 movies, and thus we could be interested in their relative spacing, Fig. 8.1 does not give us enough information to make good judgements. There are certainly more top 250 movies as time goes on, but is this simply because more movies are being produced? We cannot tell from the available data, and so the relative spacing is less important. If we want to see if movies are becoming "better" or "worse" over time (at least according to popular ratings), keeping a linear mapping of time does not help.

"Now" is often a good candidate for a focus point for a distortion of time. Focusing on the present and giving more visual space to the immediate is a good choice since action is more likely to be based on current, rather than historical, data. This notion of distorting to show the immediate can be seen in more than just time representations – *The New Yorker* magazine of March 29, 1976 has a wonderful map showing a picture of the world as seen from 9th Avenue in New York City. Although meant somewhat ironically, if your goal is to present information to be of use, then

[1]Of course, different data would change this. If we were looking at the weekly box-office grosses, as we do in Fig. 5.9 on page 115, then the duration of a movie becomes a useful datum. The data drive the visualization.

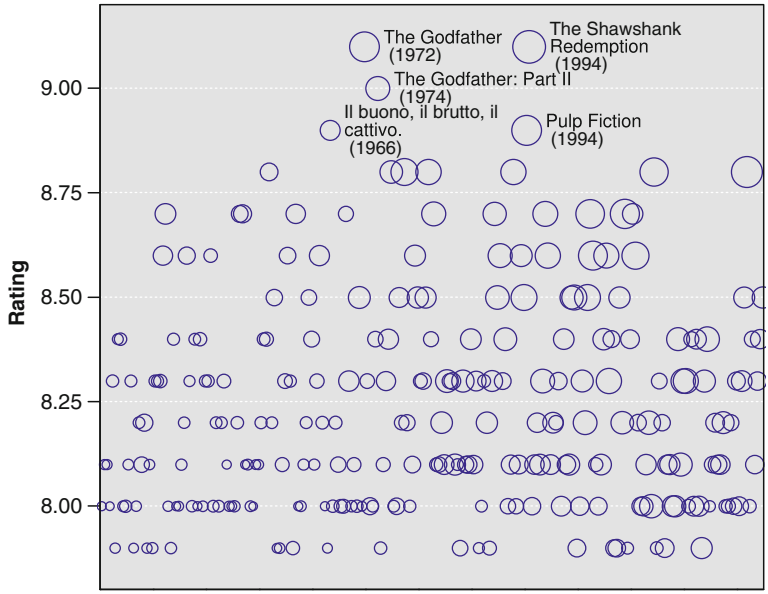


Fig. 8.2 Top 250 movies as of January 2010. This chart shows the highest rated movies as a scatterplot of ratings by time. The number of votes for each movie has been mapped to the size of the points. In this version of the figure the dates are shown as an ordered list, with the *x* dimension simply showing the order within the data set

if you live or work on 9th Avenue, it makes plenty of sense to give lots of detail of that street and to have everything past the Hudson River reduced to a single icon or less.

Figure 8.2 shows a first distortion attempt that can be done without any special knowledge of the data. It simply replaces the dates by an ordering of the dates – so the *x* dimension simply represents the order in which the movies were released. This ensures that the events do not bunch up – in fact it spreads them apart as much as is possible and so allows us a good view of the order of movie ratings. It looks like there was a peak in the middle, but not a terribly strong one. Some form of confirmatory statistical analysis would be needed to see if it was "real."

One problem with this figure is that time has become demetricized – we cannot tell without more detail if a range on the *x* dimension represents a short or a long time, and since the transform from time to sequence depends entirely on the data, if we changed the data (say, to include the top 500 movies), we might have a very different mapping.

Figure 8.3 is an attempt to get the best of both worlds. Noting that the present is a good focus point and that the data are more dense at this point, a square root transform has been applied, making the positions along the *x* axis proportional to the square root of the time between that date and the present. The result is similar to Fig. 8.3, but now the horizontal dimension has a fixed mapping and varying the

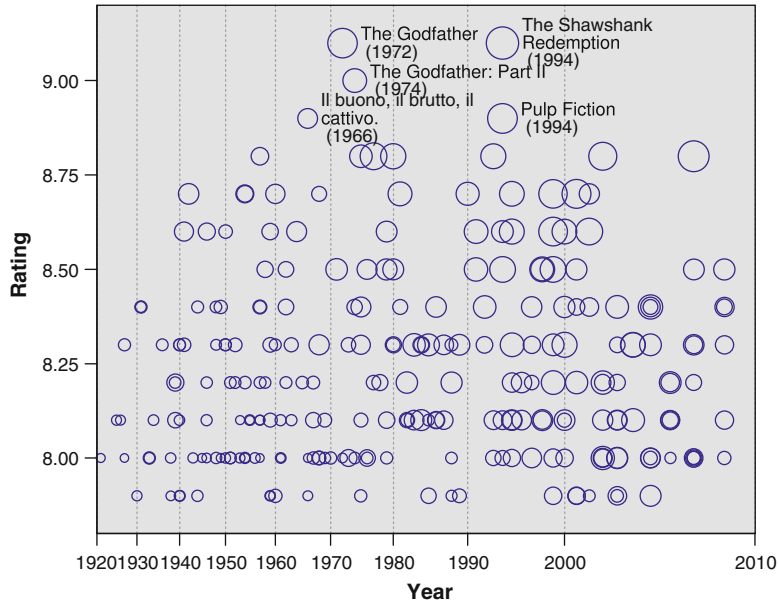


Fig. 8.3 Top 250 movies as of January 2010. This chart shows the highest rated movies as a scatterplot of ratings by time. The number of votes for each movie has been mapped to the size of the points. A square root transform has been applied to both the x dimension and the size aesthetic. For the size, this is so that the area represents the counts. For the dimension the purpose is different; the x dimension has been transformed so that more recent movies are given more space to be displayed – the time scale has been distorted to focus attention on the important time period

data will not change the positional information. We can also use gridlines as a rough form of binning into decades, so crude judgements about relative densities can be made. For those of us who spent much time at the cinema in the 1980s, we can also see what we sadly suspected – it really was a lean time for good cinema.

8.2 Time as Frequency

When a series of values in time is observed, it is often possible to see repetitive patterns in the data. Sometimes the presence of periodic patterns is suspected, but not as obvious. Figure 8.4, for example, shows a time series for each of three components of the UK's consumer price index.

In the lower figure, there appears to be some form of periodicity in the data, whereas in the top two time series we cannot see that feature as clearly – if it exists at all. However, based on our knowledge of the data, we might expect to see patterns showing the cyclical nature of the data. It seems natural to hypothesize that food prices have a seasonal component, for example.

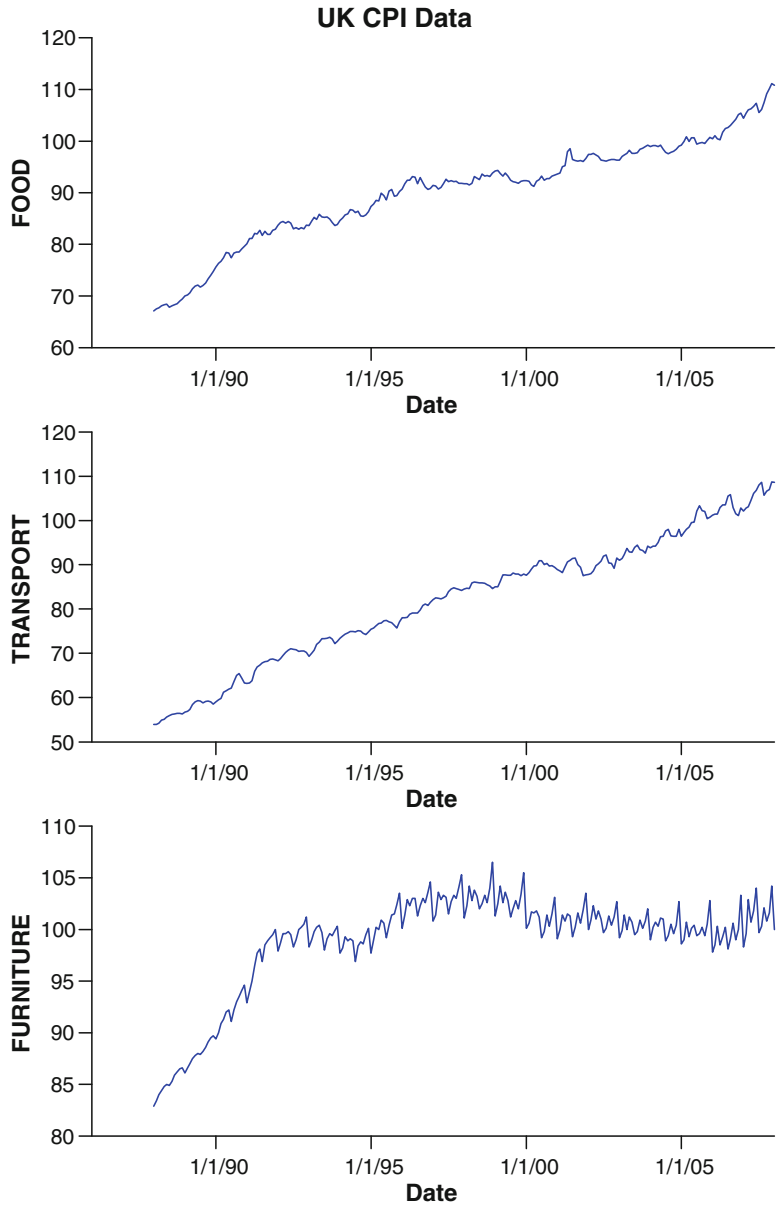


Fig. 8.4 Time series line plots of consumer price indices for the UK over the time period 1988–2007: *Top to bottom* food, transport, and furniture prices

In order to show periodicity more clearly, one technique is to radically transform the data; replacing the horizontal dimension of *time* with one of *frequency*. This type of analysis is called *spectral or frequency analysis*. All books on statistical time series analysis will contain a section on analysis of the frequency domain or spectral analysis. Chatfield's book [17] is one of the most read introductory references, and the reader is directed to this and other similar texts for details on spectral analysis; the general approach has been used since before the 1900s, for example, as described in [96].

The basic concept is to decompose a time series into a sum of sinusoidal components and to estimate the coefficients of those components. It is strongly related to *Fourier analysis*, and indeed the most basic decomposition used in spectral analysis is the finite Fourier series representation. For the series $\{x_t\}$ of length N, this decomposition is given as

$$x_t = a_0 + \sum_{p=1}^{p=(N/2)-1} \left\{ a_p \cos\left(\frac{2\pi pt}{N}\right) + b_p \sin\left(\frac{2\pi pt}{N}\right) \right\} + a_{N/2} \cos \pi t,$$

and the coefficients are easy to calculate directly from the data. The coefficients then give a measure of the periodicity of the data; the term $R_p = \sqrt{a_p^2 + b_p^2}$ is called the amplitude of the pth harmonic, and it measures the strength of that harmonic – if it is high, then there is a strong periodic component with that frequency. In fact it can be shown that R_p^2 is proportional to the contribution of the pth harmonic to the overall variance of the data, so that these values decompose the variance.

A plot that shows R_p^2 plotted against p is termed a *periodogram*,[2] and it can be used to examine a time series for periodic effects.

Figure 8.5 shows periodograms for the same data as given in Fig. 8.4. The periodicity we saw in the Furniture data is confirmed – in fact we can see that the strongest high-frequency terms are those for two and four cycles a year – stronger than the yearly cycle. This is an unusual feature for what we might expect to be seasonal data. The middle visualization shows Transport, and that is more similar to what we might expect, with a peak at the yearly frequency mark. Food, perhaps surprisingly, does not appear to have much of a seasonal effect, with low-frequency (long-term) effects dominating.

This data set is analyzed for periodicity in a more direct way in Sect. 9.2.4.1 on page 192, where we use a more primitive technique of fitting one frequency at a time and seeing how well each one works. This is the way frequency analysis was performed prior to 1900, and although the periodogram has advantages in being able to show multiple frequencies all at once, the value of directly estimating periodicity and showing the differences between the fit and the actual values is worth noting.

[2] Although, since it shows frequency on the horizontal dimension, a more accurate term would be "spectogram," as some authors indicate. Also note that the value plotted is typically $NR_p^2/4\pi$ rather than simply R_p^2.

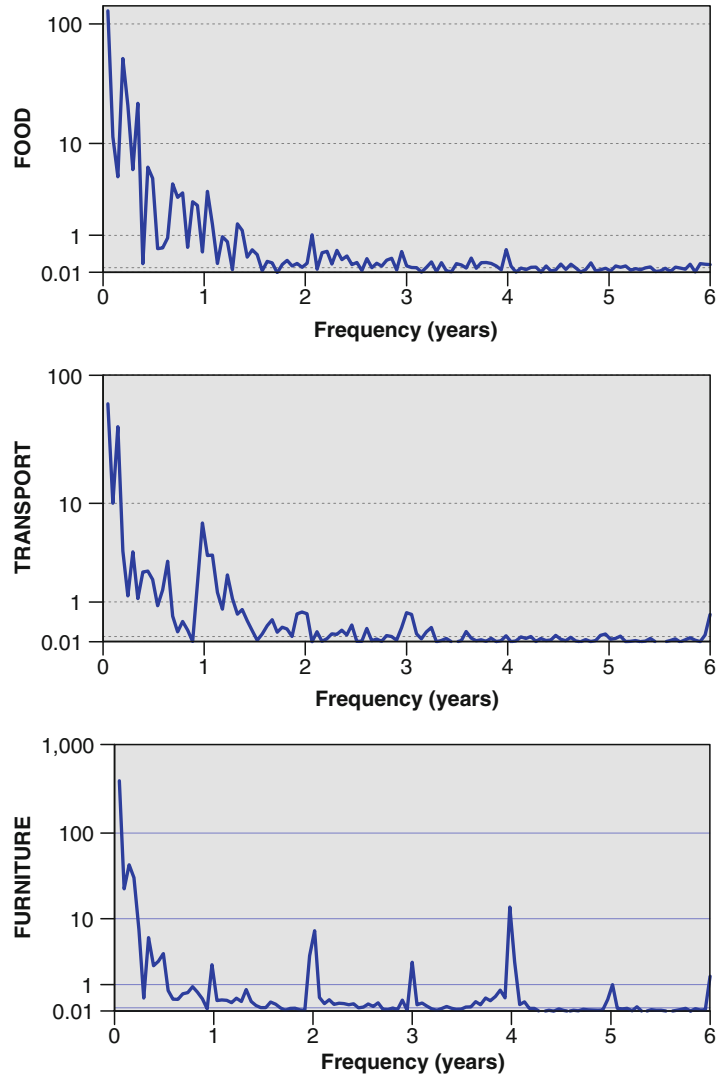


Fig. 8.5 Periodograms for UK CPI data. These three figures show periodograms for three different types of consumer price index in the UK over the time period 1988–2007. *Top to bottom*: food, transport, and furniture. The data have been detrended by fitting a simple linear trend, and the resulting residuals have had spectral analysis performed. The y axis is shown on a log scale to emphasize the smaller contributions

8.3 Converting Between Categorical and Continuous

8.3.1 From Categories to Continuous

Chapter 4 gives a number of ways in which we can consider a time variable, with the most important choice being whether to treat time as a *continuous* or a *categorical* variable. Since time is intrinsically continuous, it is relatively easy to convert from a categorical time variable to a continuous one. The only difficulty arises when the variable represents a range of times. For example, we might have stock prices summarized over a time period, such as shown in Fig. 8.6.

Plotting these data is more challenging than we might expect. If we just connect the opening prices with a simple line, it hides the fact that the opening price applies to a whole month and gives the impression that the value being displayed changes smoothly, whereas in fact it really is just a single value that remains unchanged for the month. In Fig. 8.6, the line displayed is a *step line*, a form of interpolation that keeps the value constant between observations. This version of the step line keeps the value the same until a new observation is encountered, which is what we need for the opening price (the opening price is the same for the following month). The model here is that in the absence of data, we assume that the price stayed the same until the next observation.

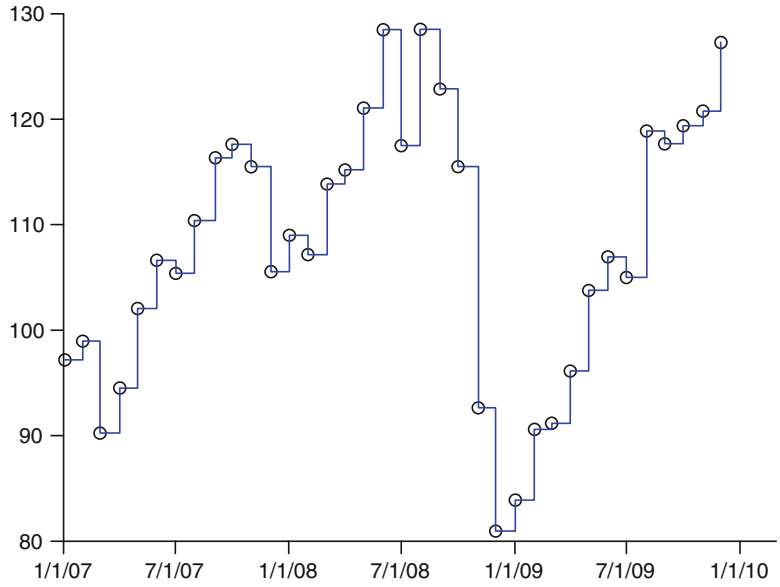


Fig. 8.6 Price of IBM stock. This data set contains monthly data on IBM stock for several years. Of the available variables that summarize the price – open, close, high, low, volume – this chart plots the opening price for the month. The *points* show the data, and the *step line* is used to show that the opening price applies to the month

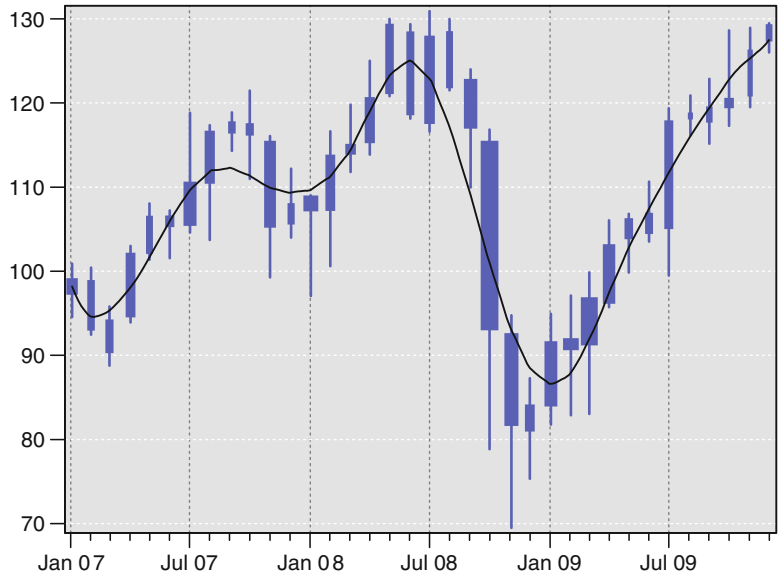


Fig. 8.7 IBM stock. This data set contains monthly data on IBM stock for several years. One edge element links the high and low prices for the month (*thin line*). Another edge element links the open and close prices, shown as a *thicker line*. The width of that *thicker line* denotes the volume. A *smooth line* is an explicit estimate for the actual price, smoothing the start and end prices for the data

The open and close observations are really point values in the data – although assigned to a month, the opening price represents the price at the very start of that month and the closing price the price at the very end.[3] Other variables in this data set include high, low, and volume. The first two have a definite time, but that time is unknown – it is simply "sometime that month." The last is an aggregate summed over the whole month. For these fields, if we need to place them within a continuous time dimension, then using the center point of the time period is a reasonable choice.

As a general rule, it is better to keep the distinction between continuous and categorical time obvious by using multiple elements, as in Fig. 8.7. The monthly summaries have been displayed using elements that are disjoint – they clearly show that they represent individual months; no confusion exists as to what time they denote within the month, and there are no issues of interpreting a line joining months.

However, there is an underlying process that is more continuous, and a line element is a good way to display that process. Since the data cannot give us that information, we know that we will end up making an estimate, so we might as well

[3]It is important to note that the price at the end of one month is not necessarily the price at the beginning of the next month, since there is a gap between the two times where the stock is not being traded. It is therefore common for the price to be different. In fact, if the company releases financial information outside of normal trading hours (as companies typically do), there is very likely to be a large difference.

use something more informative than simply joining up values. In Fig. 8.7 we have added a smooth line that gives us a visual indication of how the stock has been trending. It is clearly a smooth (note that for 2 months it gives an estimate that is outside the range for that month), and so this is a *truthful* chart: It represents the data it displays faithfully.

8.3.2 From Continuous to Categories

A standard way to convert data from a continuous domain to a set of (ordered) categories is by binning. There are a variety of ways of doing this, but they all share the same basic concept: A set of extents are created that completely cover the data being binned. All data within an extent are summarized by a set of values that represent that extent.

A simple form of binning is that used in simple histograms. It is one-dimensional – the extents are simple ranges. The bins are all the same size; they do not overlap, and they all touch each other. We can define the extents as follows:

$$[\alpha + i\delta, \alpha + (i+1)\delta) \qquad \forall i \in \{\ldots -2, -1, 0, 1, 2 \ldots\}$$

In practice, only those bins that cover the data spread are retained (data storage for an infinite set of extents tends to be expensive, as well as pointless). So for this simple case, all we need to define is a bin width δ and an offset α. The choice of δ is, however, not easy for an automatic program. Simple rules are possible, such as creating a number of bins equal to $|\sqrt{N}|$ or $|1 + log_2(n)|$ (Sturges' formula) when there are N data points. They have the benefit of simplicity, but they often work poorly for small numbers of N and, worse, tend to create bad bins when the data are *granular* – that is, the recorded values of the data are recorded to a certain precision or for some other reason have a minimum difference. This is very often the case for time data; they are recorded at consistent intervals or at a certain precision.

A good algorithm takes the granularity into account (so that weekly data will not get bin sizes of 10 days) and chooses bin sizes that fit the type of data, so that time-based data get interpretable ranges – minutes, hours, days, weeks, months, etc., and if multiples are used, then multiples that make sense are used (5 minutes and 10 minutes are good multiples, but so are 15). Seven days is also reasonable, whereas 7 centimeters would not make sense for data that represents lengths). The material in Chap. 6 on creating tick marks for axes is also appropriate here, with modifications to allow for a variable desired number of divisions.

Many programs for dealing with data include functions for transforming time; these can be used as aggregation methods when you have an idea of the sort of bins you want to create. Table 8.1 gives a set of functions that can be used in Excel to aggregate times. Care must often be taken to ensure that results are as expected – there is no standard mapping of day of the week, for example, so in the expressions needed in Excel, parameters have been set to ensure that everything is one-based.

Table 8.1 Useful Excel formulas for transforming dates. In the formulas below, the indices returned are one-based in all cases. They sort correctly and the formulas respect leap years (e.g., the "fraction of a year" formula calculates the length of the year in days rather than assuming 365 days in a year)

Transformation	Formula	Result
Year	YEAR(A1)	Year with century
Month of year	MONTH(A1)	{1 . . . 12}
Day of month	DAY(A1) *	{1 . . . 31}
Quarter	CONCATENATE("Q", (INT(MONTH(A1)/4))+1)	{Q1, Q2, Q3, Q4}
Day of week	WEEKDAY(A1,2)	{1 . . . 7}, 1 = Monday
Weekend	IF(WEEKDAY(A1,2) > 5, TRUE, FALSE)	{false, true}
Day of year	A1-DATE(YEAR(A1),1,1)+1	{1 . . . 366}
Fraction of year	(A1-DATE(YEAR(A1),1,1)) / (DATE(YEAR(A1),12,31)-DATE(YEAR(A1),1,1))	[0 . . . 1]

Another helpful hint is to choose a mapping such that weekends appear together. Very often the work week has different behavior from weekends, and so it is better to see Saturday and Sunday beside each other rather than at opposite ends of the week, as is the traditional method.[4]

By taking combinations of the expressions in Table 8.1 various time aggregations can be made that are interpretable and so can more easily be used to make decisions from.

8.4 Summary

Time data are inherently linear, continuous, and directional. This chapter has demonstrated ways in which those fundamental properties can be sidestepped – distorting the linear nature of time to provide emphasis or focus; binning time to make it discrete; using frequency analysis to consider time as a a cyclical dimension.

Distortions of time allow us to fast-forward through times that are less important and focus on the important parts. It is like watching an 8-hour cricket match in

[4]Judaism, with Christianity following it, has always firmly placed Sunday as the first day of the week. The Sabbath (Saturday) is the day when God rested at the end of the week of creation. So with Saturday the last day, Sunday must be the first. On 7 March 321, Constantine I, Rome's first Christian emperor, decreed that Sunday would be observed as the Roman day of rest and formally abolished the previous 8-day week (which no one was using anyway). From then until the modern era, most of the Western world has tended to start the week with Sunday. However, when Saints Cyril and Methodius brought Christianity to the Slavs, they, for some unknown reason, decided to start the week with Monday, so those countries are an exception. On the other hand, the term *weekend*, which is first recorded around 1880, places Sunday at the end of the week. Modern business use typically follows that concept, starting the week with Monday. The International Standards Organization has defined the week as starting on Monday [63], so it does seem as if Monday is winning. Maybe it's time for me to change that setting in iCal . . .

a few minutes, zooming through the routine parts and slowing down for the big hits, catches, and bowling successes. Just as we rarely have time to watch sports for 8 hours, we often find we cannot represent a large time expanse on a page or a computer screen, and distortion techniques allow us to see what is most important.

Frequency domain analysis is outside the scope of this chapter. The basic concept is to allow us to see cyclical effects by highlighting similarities at different frequencies. Discovering seasonal patterns is an important finding, as it allows us to understand the subject more easily as well as allowing us to build better models to predict the future.

Binning is used in many other places in visualization – faceting, for example, as examined in Sect. 6.4.4 on page 148, or when we need to use a variable for an aesthetic that produces discrete outputs (Sect. 7.1 on page 151). It is a fundamental technique that is useful throughout visualization – even when creating simple tables a good method for "rolling up" data to an aggregated form is valuable.

8.5 Further Exploration

Chatfield's *The Analysis of Time Series* [17] is an excellent and much-followed introduction to the statistical science of time series analysis. Other similar books can be found on Amazon's best-seller list with a search for "time series analysis." To a large extent, which book to choose depends on your focus – and if you want programming examples included.

The collection of articles on information visualization titled *Readings in Information Visualization: Using Vision to Think* [16] contains a section on *focus+context* methods among a wealth of interesting papers.

The choice of width for binning is a fun aside to research. Silverman's *Density Estimation for Statistics and Data Analysis* [99] provides a standard reference. The Wikipedia entry on histograms provides some examples of commonly used formulae, but probably the best source of material on what makes good bin widths for time data is to study plots in newspapers and see examples of handcrafted solutions.

Chapter 9
Interactivity

You are wrong to say that we cannot move about in Time. For instance, if I am recalling an incident very vividly I go back to the instant of its occurrence: I become absent-minded, as you say. I jump back for a moment. Of course we have no means of staying back for any length of Time, any more than a savage or an animal has of staying six feet above the ground. But a civilized man is better off than the savage in this respect. He can go up against gravitation in a balloon, and why should he not hope that ultimately he may be able to stop or accelerate his drift along the Time-Dimension, or even turn about and travel the other way?

— H.G. Wells, *The Time Machine* (1898)

9.1 A Framework for Interactivity

There has not been a wealth of published material on systematic approaches to interactions with graphics. Cleveland and McGill's book *Dynamic Graphics* [23] provides an early guidance, describing sets of techniques for interaction and display of the results of interaction that guided much research. Like other subsequent books, it does not attempt to provide an overall framework but rather describes a set of methods and their utility. Often, interactivity is discussed only in relation to a certain style of plot, making it hard to understand when techniques might be generally applicable and when they are specific to a given plot. For researchers implementing interactive systems, this gives little insight into the tradeoffs between a flexible interactive system and an efficient one. In this section I will propose a general theory of interaction that is based on a display pipeline that is typical of systems that draw static graphics. The pipeline is augmented to show how interactivity can be added to the static display of data. This approach fits in with the general grammatical

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_9,

basis that this book uses to describe graphical representations as in this formulation; interactivity is simply another orthogonal component that can be used in conjunction with the other components of the grammatical description of visualization.

That is not to say that all approaches are useful with all plots. There is little point, for example, in allowing 3-D interactive rotation for a simple 2-D chart (even though it is possible to do so). More importantly, different techniques are useful for different forms of data. In this chapter interactions that are of particular interest in the visualization of time data are given prominence.

9.1.1 Display Pipeline

When drawing a graph of data, a system starts by accessing the data and ends by producing a visual display. What happens in between is controlled by settings or parameters. Thus the basic equation of visualization display is

$$data + parameters = graph.$$

Interactive graphics are graphics that allow the user to manipulate one of the two inputs, data or parameters, and show the changes in the chart. Even a very simple visualization system such as Microsoft Excel allows users to adjust some settings on the transition from data to display. The more detailed the available settings, the more powerful interactions can be defined. Modifying data is also a universally available technique, as a visualization system that does not let you choose what data to view would not be very useful. The basic approach to adding interactivity to a visualization engine is therefore simple:

- Allow the user some graphic interface mechanism to change data or set parameters.
- Re-execute the pipeline and replace the original display with the new one.

In early applications (e.g., PRIM-9 [40]) this naïve approach was simply too slow; special code was written to make sure that updating displays could be done quickly and only parts of the pipeline re-executed. On modern computer systems this simple approach is now fast enough so that it often works well enough without the need for optimization. In those cases where it does not work well enough, care needs to be taken to execute only the parts of the data pipeline that need to be executed. These cases are often when visualizations include

- Large amounts of data;
- Slow access to data;
- Complex, and especially iterative, calculations;
- Complex output, such as 3-D scene trees.

Figure 9.1 shows the basic pipeline. It omits components of charts such as axes, legends, and titles as interacting with them is less of a data visualization exercise

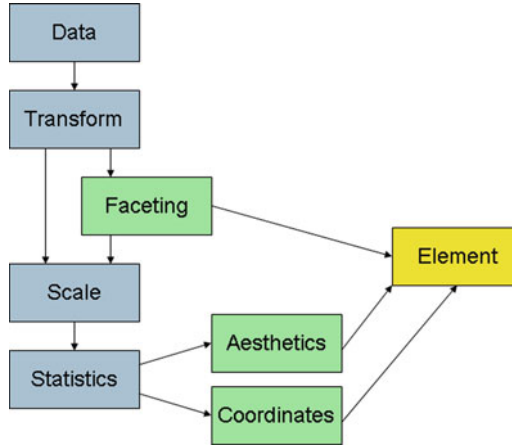


Fig. 9.1 Display pipeline: data to element. This figure shows the steps in the processing chain that starts with the data and ends with the graphic element. The steps are defined using the terms of Chap. 2, but the basic flow is universal to any system for building charts. In the simplest chart, such as an unadorned scatterplot, we may only see a few pipeline steps (data, coordinates, and element), and it is possible to inject more complexity into the diagram, but by and large, most systems' underlying architecture will look similar

and more of a graph presentation exercise.[1] Good axes and legends are important, but modifying them leads to clearer presentations rather than aiding exploration – a laudable goal, but not one that is as valuable for the interactive exploration of data.

Data are read in and transformed as desired. If the data are to be faceted into more than one panel, that is done before scale and statistic calculations as those may need to have different results per panel (for example, a statistic that cumulates values over time should cumulate separately within each panel). Scales (such as log or power scales) can be applied to variables, and then statistics are calculated on the data, if desired. The results of the statistics are used to create aesthetics and for the coordinates (locations) of graphic elements with each panel. At each stage of this pipeline there are parameters that control how the pipeline operates, and data can be injected into several of the steps; the following sections discuss how this can usefully be achieved.

One important point to note for anyone architecting an interactive graphic system is that an interactive technique that affects a step in the chain nearer the end can be made more efficient, as any steps that preceded that step do not need re-executing. For example, 3-D rotation only needs the coordinates and element rendering steps re-executed. Complex statistical calculations such as smooth surface creation can be left unchanged.

[1] *Exploration* is characterized as being *fast*, *immersive*, and *data-focused*. *Presentation* is characterized as being *precise*, *reflective*, and *view-focused*.

9.2 Modifying Parameters

The simplest interactive technique is also the most powerful; virtually any chart is specified using a number of parameters – the numbers and values that define details of a display, such as the number of bars in a histogram, the power function on a transform, or the projection angle in 3-D or higher-dimensional projection techniques. Putting these parameters under interactive control allows users of a visualization to explore details of the view, validating that the view is truthful, and checking for a better representation. If the view involves any form of modeling, then modifying that parameter is a valuable tool for exploring how good the model is at explaining the data. This section details how parameter manipulation can be applied at each step of the display chain.

9.2.1 Modifying Element Parameters

Working backwards through the chain, the element-rendering step has relatively few parameters that can be usefully interacted with. It would be possible to modify 3-D lighting models or allow interactive control over how to display missing values in time series, but to a great extent these are not interactions aimed at understanding the data. Given coordinates telling it where to draw and aesthetics telling it how to draw, the element has few choices that need to be made. One property that *can* be set is whether or not the element is visible at all. This technique is of less use when there is only one element, but with multiple elements it can be useful to have always visible one element that summarizes the overall data and another element that shows details only available at a certain level of magnification, or only in a region that can be moved over the plot by the user. This is an example of a *focus+context* plot, a display that shows the overall data (the context) together with details on part of the data that the user has indicated (the focus). Although originally introduced as a distortion technique (which we will demonstrate as a coordinate technique in Sect. 9.2.3), with a good survey paper by Leung and Apperley [71], the term is more widely applicable, with Card [16] providing a definition that highlights the generality of the approach.

Figure 9.2 shows an example interaction that selectively shows one element to provide a focus+context enhancement to a chart containing a single base element; the chart shows one element (an interval element) all the time and a second element (labeled points) only in a magnified region. The "magnifying glass" region can be dragged around the plot, making it an interactive control.

The data for this plot consist of information on emails sent and received by the author during 2007. Emails contained data on when they were sent (both date and time of day), the sender and the receiver, the subject, the contents, the size of the email (including attachments), and the recorded importance of the email. The first step in the analysis of these data was to simplify the text content. For subject, this was achieved with the following procedure:

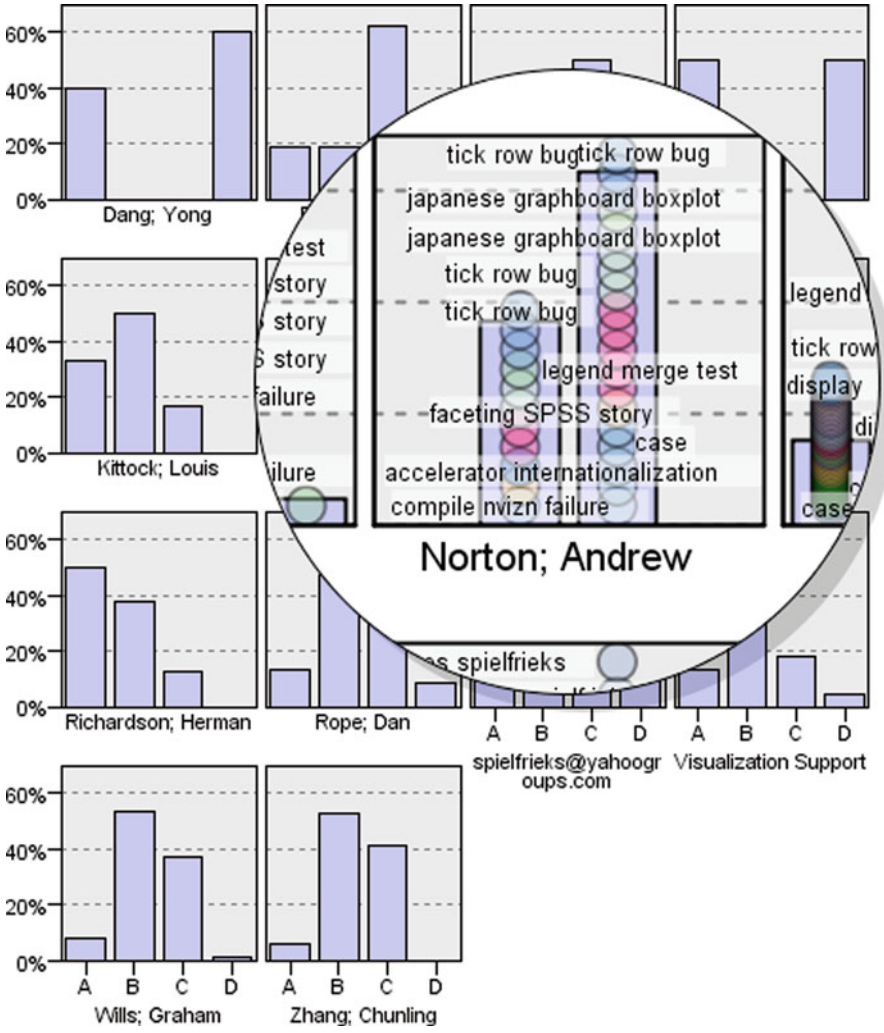


Fig. 9.2 Email items sent and received in 2007. Email data are displayed as two elements, one for context and a second for detail. The detail element is only shown within a region defined by the user, using mouse gestures to drag it over the chart and focus on areas of interest

- All words across all subjects were recorded, along with their frequencies.
- "Stop list" words (words such as "of," "and," "the," and so on that are not informative) were removed from the list.
- The 200 most commonly occurring words were retained and an indicator variable created for each word that identified whether the keyword occurred in the subject of each email.
- A cluster algorithm (k-means) was run on these 200 variables to cluster the subjects into a small number of groups. Emails that were only weakly assigned

to a group were dropped from the set to be considered, leaving only a small set of emails divided into distinct groups.
- The three most frequent keywords found in the group were used as the group name. Some groups did not have three such keywords and were named with shorter names.

For contents the first three steps were performed, but the contents were not clustered. Instead, the contents were simply reduced to the five most frequent words occurring from a candidate set of 500 commonly occurring words, resulting in many different variable values.

In Fig. 9.2 the time component has been transformed into four time intervals, with "A" being the hours from midnight to 6 a.m., "B" being from 6 a.m. to noon, etc. The chart has been faceted with each panel displaying the sender of the email and the contents of each facet being a summary of the percentage count of emails sent during each time block. The facets have simply been wrapped inside the display area rather than showing as a single (long!) row or column.

Thus the first panel shows when Dang Yong sent email to the author, which is fairly evenly split – either before 6 a.m. or after 6 p.m. That time pattern is the result of geographical situation; Dang Yong works in Xi'an, China, while the author works in Chicago.

One major problem with percentage summaries is that they hide the actual counts. We cannot tell if Dang Yong sent many emails or few from this view. For this reason the view has been augmented to show a second element, consisting of one dot per email, with color and label redundantly encoding the subject of the email. If shown for all panels, this would result in a cluttered chart, making it hard to understand. Instead, a *magnifying-glass* interaction has been added. The detail element has been hidden in the overall chart, but within a circular region around the mouse location both elements have been displayed, and displayed at a higher level of magnification. In Fig. 9.2, the mouse has been dragged over the center of the facet corresponding to emails originating from Andy Norton. He sent email during daytime hours only, and we can see both how many emails were sent and their subjects.

As we move the mouse over the view, we can explore the details on the focus locations while retaining an overall context. Since this method simply selectively reveals information on the elements, it is a very fast technique, requiring only rendering of elements and no pipeline re-execution. This is in contrast to other focus+context methods such as the fisheye transform of Sect. 9.2.3.

9.2.2 Modifying Aesthetic Parameters

Setting parameters on the aesthetics allows you to interactively modify the mapping from data to appearance. An example of this can be found in many image-editing packages. In Adobe Photoshop™, the "Levels" controllers allow users to adjust

the mapping between original color values and output color values. By adjusting the ranges and control points, users can emphasize differences in shadows at the expense of highlights, or vice versa, or define more complex mappings that take into account hue or saturation or values of an individual color channel. In a similar fashion, interactive control of the mapping from data to an aesthetic allows users to emphasize different ranges in their data.

Possibly the interactive feature people are most familiar with is the pop-up, or tool tip. This is a modification of the labeling aesthetic. Instead of having variables used for labels, the values are instead attached to the graphical elements as metadata and are only displayed when the mouse is moved over the element, or clicked on, or by some other interaction technique. As well as being considered a parameter interaction technique, with the parameter being the item for which to show the label, this could also be considered a data augmentation technique, since metadata are being added as an aesthetic and then selectively displayed.

The technique is simple, but it is valuable for the following reasons:

- It is ubiquitous. Since every chart contains at least one graphic element, this technique can be used anywhere. The only caveat is that if a graphic element aggregates several rows, such as occurs in a bar chart, then the pop-up needs to have a summary function defined if the values for the pop-up need aggregating. Often this is desired behavior anyway, such as when we have a pop-up for a bar chart of mean values that gives the count of the number of rows that have been aggregated into each interval.
- It provides details on demand. Because the information is hidden, it does not clutter up the basic display, allowing an overall view of the data to be seen. It is only when the user indicates that one part is of strong interest that details are made available. Suppose, for example, we had a simple time series chart showing, for each hour, the three stocks that had the largest change in value, with the y dimension showing that change. The information on which stock moved is clearly of value, but any aesthetic coding, like color, would contain too many categories to be useful, while labeling every point would result in a plot with so many labels that the data points would be indistinguishable. The solution is to use a pop-up for the stock name, as the resulting chart allows a user to see a pattern and then rapidly query for additional information. It does not overload the chart initially but makes the information available on demand. It is a simple example of a focus+context technique. As a general principle, if you have a variable that looks like a name or identifier for each row, and it has many values, a pop-up is an excellent way to add that variable into a chart as an aesthetic.
- It is well understood. Under the name “tool-tip,” the user interface for pop-ups is well known to most computer users. Basic productivity applications and operating systems typically display some helpful information if the cursor is moved over a graphic representation of an icon or other user interface affordance. A scatterplot with pop-ups giving details on each point is a readily understood system for users.

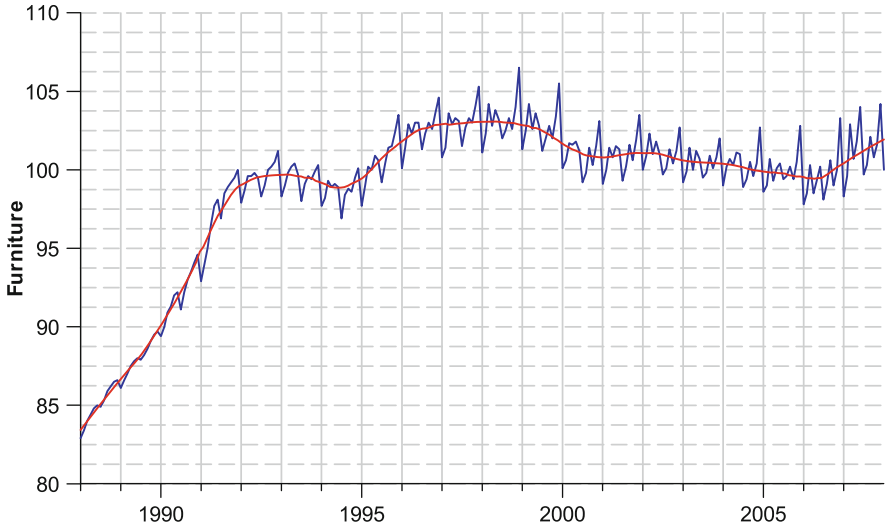


Fig. 9.3 Furniture CPI, 1988–2007. A loess smooth has been fitted to show trend changes

9.2.3 Modifying Coordinate Parameters

A common form of interactivity is 3-D rotation. Early uses of rotation for data analysis are listed in [27] – applying rotation to points to form interactive spinning 3-D scatterplots. Rotating scatterplots also have an important place as being among the first techniques to be made available on desktop computer systems, with MacSpin [34] being available on desktop machines in 1988. One of the reasons that rotating 3-D scatterplots were an early interactive invention was the simplicity of their pipeline – as already mentioned, a 2-D scatterplot has few display pipeline steps; adding a 3-D coordinate projection adds a single step relatively late in the chain.

Control of 3-D rotations can be achieved by setting the three angle parameters based on users' directly manipulating the chart (clicking on the point cloud and then "throwing" it to start it spinning), or indirectly by dragging sliders and similar controls, still an important application. This concept generalizes well to other forms of coordinate transformations. For example, in a spherical map projection of geographic data, a single point on the Earth's surface defines the center of the projection. We can therefore easily create an interactive tool that drags a point around, which in turn controls the map projection. For an example of a particular application to time series, consider Fig. 9.3 above, where we show consumer price index (CPI) data from the UK. The data consist of the UK CPI for furniture measured monthly, with January 1995 defined as the date where all indices are 100.

The data have a loess smooth drawn on the original time series (we will return to this smoother in Sect. 9.2.4). The problem with this chart is that it is hard to see the details of what is going on in the interesting area because the area of greatest interest is compressed in a narrow band about the smooth line. To remedy this we

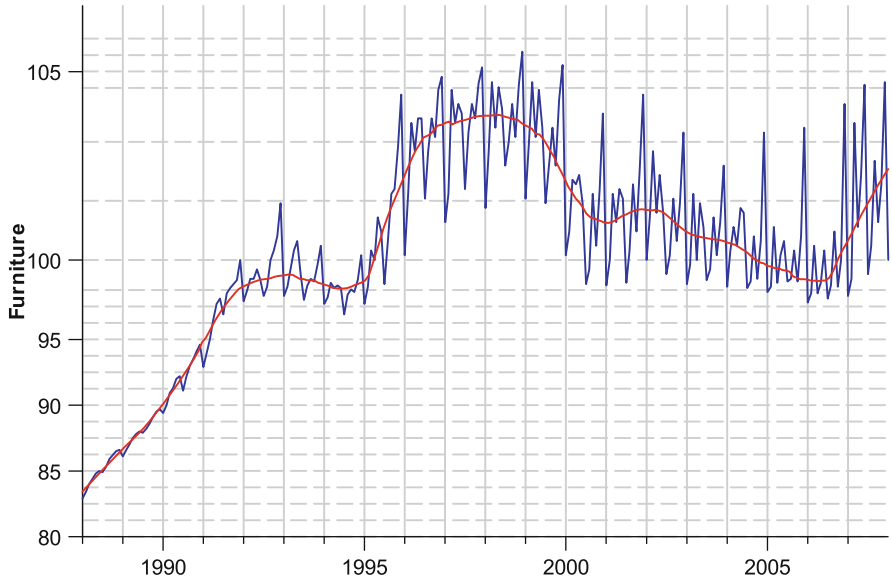


Fig. 9.4 Furniture CPI, 1988–2007. A loess smooth has been fitted to show trend changes, and a distorting transform exaggerates the data around the *line* $y = 102$

add a focus+context transformation that magnifies locally about a center point. In Fig. 9.4 we magnify the y dimension locally by a factor of 3 and drag the focus point to where it centers on the y value `102`.

The spacing between the gridlines provides a visual indicator of the degree of magnification. With this magnification, the amount of smoothing is clearer and we can see that at the bump around 1998, most data points lie above the smoothed line, with only a couple of values lying below the smooth each year. In Fig. 9.5 on the following page we magnify the x dimension by the same amount, centering on the value of 1995.

This is an effective technique for time series, as we do often want to focus attention on a small time slice of the overall data (here, a couple of years), while retaining the context of the overall series. This figure shows clearly that each year has dips at the end as well as in the middle.

Finally, in Fig. 9.6, we magnify both dimensions. There are a number of ways to do this. We could simply chain the two previous coordinates together, producing a result that would have a rectangular region of constant magnification in the middle, retaining the property that all gridlines would be parallel as the magnification decreases at points further away from the region. Instead, Fig. 9.6 on the following page uses a fisheye transformation, which decreases magnification radially, producing an effect similar to looking through a fisheye lens on a camera. In the central focus area, we have the same effect of constant magnification, but away from that region vertical and horizontal lines are distorted significantly. The deformation is more physically realistic, but it is not as appropriate a transformation

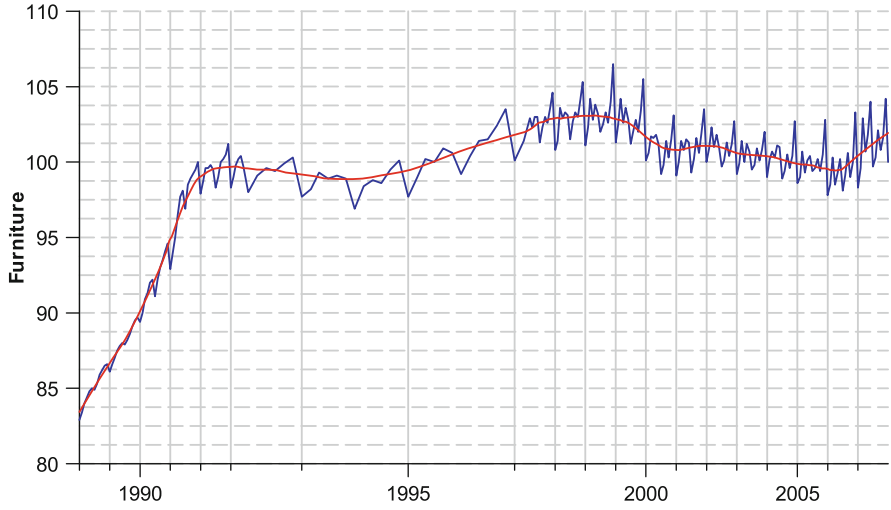


Fig. 9.5 Furniture CPI, 1988–2007. A loess smooth has been fitted to show trend changes, and a distorting transform exaggerates the data around the *line* $x = 1995$

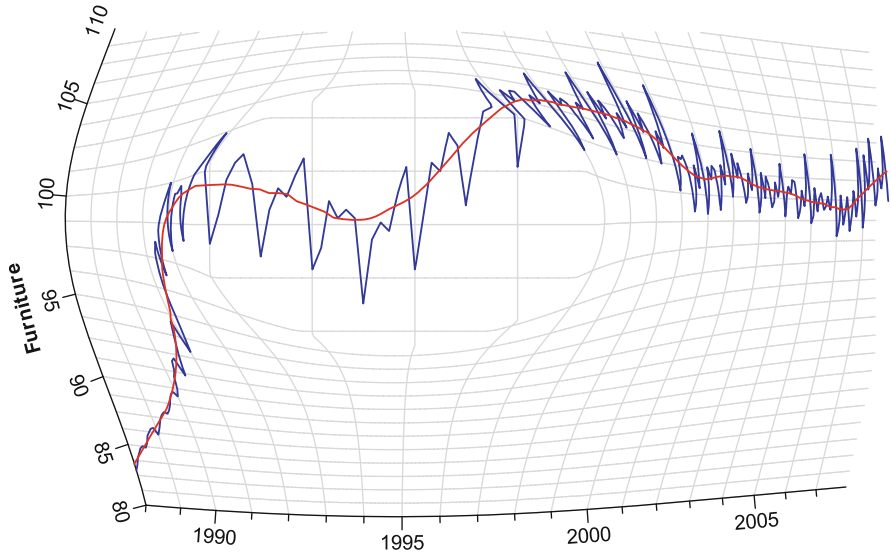


Fig. 9.6 Furniture CPI, 1988–2007. A loess smooth has been fitted to show trend changes, and a distorting transform exaggerates the data radially around the *point* $(x, y) = (1995, 102)$. This result is not a suitable one for these data, as the radial distortion combines a time dimension and an index dimension – these should be separately magnified, not magnified by one combined transform

when time is one dimension, as the radial effect "mixes" the two dimensions together, resulting in a time series line that appears to run backward. As a general principle, unless the dimensions on a chart are similar (in the sense that linear combinations make sense, which usually means they have similar units), the fisheye transformation is likely to be inappropriate, and one that handles the dimensions orthogonally is to be preferred.

9.2.4 Modifying Statistic Parameters

As anyone who has taken a course in statistics knows, there is no lack of parameters in statistics – especially those statistics that are used to build statistical graphics. The basic histogram is parameterized by the number of bins or the widths of those bins, and this parameterization is known to make a strong difference in the appearance of the histogram and, more importantly, in the interpretation of it.[2] Given the difficulties of setting a good value (and so the likelihood that any default setting is not going to be the best for your particular data set), having the ability to rapidly change the bin width is a valuable tool. Systems that feature dynamic changing of histogram bin width include MANET [116] and DataDesk [118].

Another technique that is highly parameterized is the use of density smooths, introduced in Chap. 8, where the choice of smoothing bandwidth was shown to make strong changes in the plot. Putting these parameters under interactive control allows users to check that their conclusions are not based purely on a certain parameterization and allows them to check that what they see in a chart persists when those parameters are changed. In fact the histogram *is* a form of density estimate, and so it is not surprising that the same interactive technique is applicable to both.

---

[2]There have been many attempts to automatically choose good bin widths for histograms. Scott [97] gives a formula that minimizes the mean square error between an underlying distribution and the visual representation. If σ is the standard deviation and N the sample size, then the width, W, is given by

$$W = 3.49\sigma N^{-1/3}.$$

Freedman and Diaconis suggest replacing the use of the standard deviation with the *interquartile range* – the distance covered by the "middle half" of the data or, more formally, the distance between the first and third quartile. This is somewhat more robust and gives the folowing formula:

$$W = 2\,IQR\,N^{-1/3}.$$

However, even this is not a perfect solution. If data are regular, as often happens in time series data, it is inappropriate to choose a bin width that is not a multiple of the natural spacing of the data. For example, if dates are always in days, a bin width of 1.7 days is always going to be a poor choice. For regular data, therefore, it is recommended that the optimal bin width be that integer multiple of the data granularity that is closest to Freedman and Diaconis's formula. If you want to do a truly *stellar* job, then you should also check the units of your data and bias toward reasonable bins based on that, as suggested in Sect. 8.3.2).

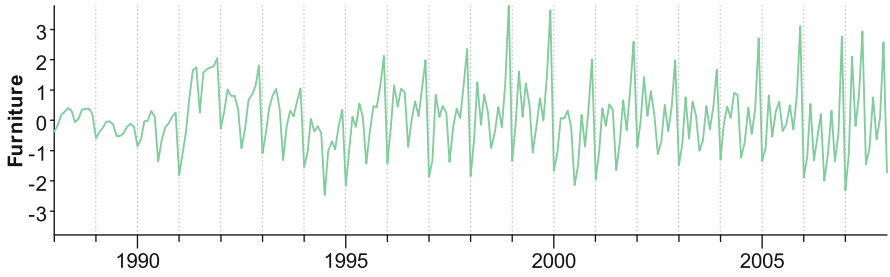


Fig. 9.7 Furniture CPI: residuals from a loess smooth with a neighborhood size of 48

9.2.4.1 Fitting Seasonality Interactively

In the analysis of time series, one common goal is to detect and model seasonality. There are many techniques that give the ability to do so, including the techniques that are described in Chap. 8. One simple way to model seasonality is to fit a seasonal model assuming a fixed seasonal effect and then put the parameter(s) that describe the seasonality under interactive control. Consider, for example, Figs. 9.3 to 9.6 on pages 188–190. In those figures we fitted a loess smooth. We used a parameterization of the loess smooth that sets the bandwidth of the smooth to include the 24 points on either side of the date being estimated. That number was chosen by adding an interactive slider to the loess fit and examining the residuals for that fit. The largest number of neighbors that gave a good overall fit was chosen. The reason to choose the largest is that, in general, we would like to have as much smoothing as possible. If a smooth has a small bandwidth, then it is only smoothing based on a small number of points around the target point and will therefore follow the data very closely – it has little "modeling power." If the smooth is too large, then it does not detrend the data well.

An example of where we have smoothed too much is shown in Fig. 9.7. This figure shows the residuals from the loess smooth plotted against time. The figure is an example of a parameterization that has oversmoothed the data; the residuals show trends between years as well as the short-cycle trends that we would expect to see within years, especially in the years 1991 through 1995 when there is a clear downward trend. When the neighbors parameter was reduced to 24, the result was satisfactory, giving the green line that is shown in each panel of Fig. 9.8. As an aside, Fig. 9.7 does not differ much if we choose values around 48 – we could as easily have chosen 40 or 50. We chose 48 because we knew this was a monthly time series and so it was helpful to be able to say that the loess smooths data at each time point by considering data from the 4 years surrounding that time point – again, as in the choice of histogram bin width, knowledge of the data makes it possible for us to use a better choice than default algorithms. This is an example of the power of interactivity when guided by statistical algorithms – the computer gives *optimal* answers in a mathematical sense, while the user can modify them to give answers that are *actionable* or *sensible*.

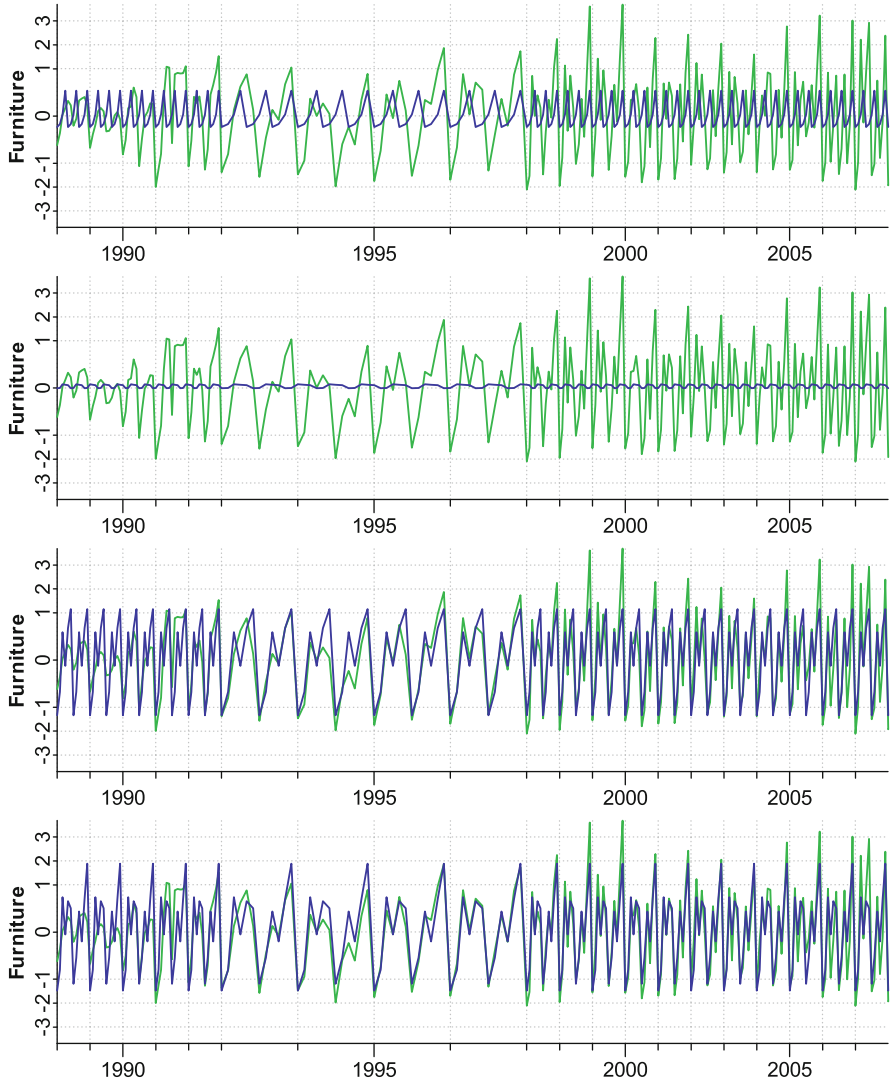


Fig. 9.8 Fitting seasonal models to furniture CPI. Each pane shows the raw data in *green*, with fitted data overlaid on top. *Top to bottom*: Fits for cycles of lengths 4, 5, 6, and 12

Having accounted for the overall trend, the next step is to discover the seasonality of the data. The input data for this are the detrended data – the residuals from the previous step. To do this a very simple model was chosen that, when given a cycle length of n, simply predicts the value at a time period i by the mean of the values at

$$\{\ldots, i-3n, i-2n, i-n, i, i+n, i+2n, i+3n, \ldots\}$$

This model has n parameters, one for each point in the cycle. If we have $n = 1$, we simply have a single mean for the whole data. If n is set to be the length of the data, our model *is* the whole data – clearly an unhelpful result. As with any model, we want to get the best fit we can with the minimum number of parameters, so we want to find the lowest value of n that provides a good result. Figure 9.8 shows the seasonal fits to the detrended data in blue, superimposed on the green detrended data. It is an easy visual task to compare the fit visually.

The top panel shows the results when $n = 4$. The peaks seem to match some of the peaks in the data, but there are regular dips that are being predicted by peaks. We do not need to see a summary statistic of this fit to know it is of little use. When $n = 5$, the situation is even worse, as the second panel shows us. The seasonal fit is of negligible size – a cycle of length 5 is of no help whatsoever. This indicates that if there is a seasonal component, its length is likely to be coprime with 5.

The remaining panels for $n = 6$ and $n = 12$ both provide a good fit to the data. The fit for $n = 12$ is not surprising. We would expect that sales of furniture would have a yearly seasonal component. In fact, were we working rapidly and not looking at many seasonal models, we might have simply fitted a yearly seasonality, seen that it was good, and moved on, missing the information that a cycle of length 6 looks good too.

In Fig. 9.8 we can compare the predictions for $n = 6$ and $n = 12$ clearly. In the plot for $n = 12$, we see that if we break it evenly into two parts, those two parts do look very similar to each other (and if we averaged them, we would get exactly the results for $n = 6$). The only real difference is in the last data point of each subcycle. In June (the sixth month), furniture sales are lower, on average, than in December (the twelfth month), and this appears to be the only real difference between the cycles of size $n = 6$ and $n = 12$.

With this detailed knowledge in mind, the final model for this time series could be comprised of the following components:

- An overall trend consisting of a smooth with a bandwidth equal to ± 2 years.
- A 6-month seasonal component.
- An adjustment factor for the increased December effect.

By graphical analysis of the smooths, varying the seasonal parameter, not only was the unexpected good fit found for half-year seasonality, but also the additional effect for the last month in the year, an effect that would not be easy to discover using traditional tools for the analysis of seasonality. It enabled us to get the goodness of fit for the 12-parameter yearly seasonality model using only 7 parameters – fits for each step in the 6-month cycle plus a single adjustment for December. This is a result that is not only parsimonious but also intriguing.

9.2.5 Modifying Scale Parameters

A simple and useful parameter interaction on scales is to use a power transform on the scale for one dimension, and then vary that parameter to see which value

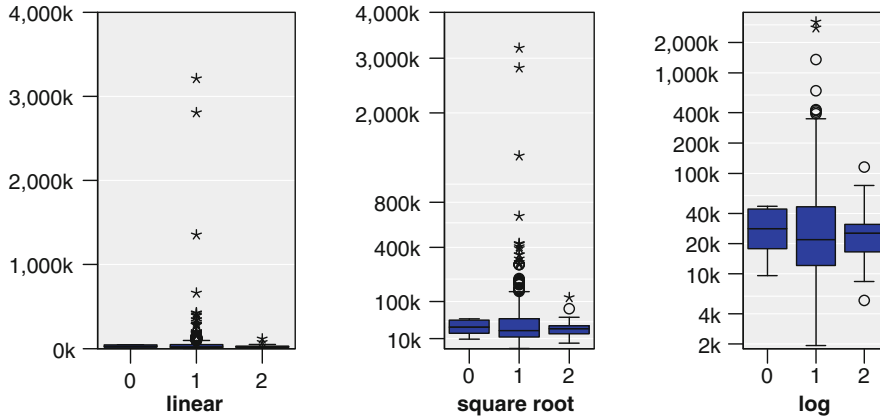


Fig. 9.9 Box plots of email sizes. The three panels show the same data, but with different scale transformations. Each plot places the importance of the email on the *horizontal axis* to form three groups. Mail that has been classified as "bulk mail" has an importance of zero, mail that has been flagged as urgent has an importance of two, and other mail has an importance of one

produces the best results. "Best" might mean many things; for a scatterplot it might mean adjusting the power scale so that the resulting point cloud looks linear, so a regression mode would be a good model. In the following example our target graph shows distributions of a continuous variable, and the goal is to minimize the skewness of the chart and choose a transformation that fits the data well based on that.

Figure 9.9 shows data on sizes of emails using the data set described in Sect. 9.2.1 on page 184. Each panel shows the results of a different value of r in the scale transformation $y = x^r$ using parameter interaction. On the left we have $r = 1$, in the center $r = 0.5$, and on the right $r \approx 0$.[3] For these data, the log transform on the right seems the best; it stabilizes the skewness so the data are more symmetric, and we can see some interesting details such as the fact that bulk mail (importance = 0) has the most constrained range, with urgent mail (importance = 2) more variable in size and regular mail the most variable. Perhaps surprisingly, there is no significant difference in the median sizes of emails in the three groups.

9.2.6 Modifying Facet Parameters

The process of faceting splits data into groups using categorical variables. This splitting is defined purely by the data, and so interactive parameter control is not

[3] If we actually set $r = 0$, then we get a flat line in the center as the transform then becomes $y = x^0$, or $y = 1$, so this graph was produced by setting $r = 0.000001$. Mathematically this function is extremely close to the log function, so for practical purposes, choosing this value of r is the same as choosing a log transformation.

generally applicable.[4] In Chap. 5 we look at an extension of simple faceting where we use statistics on the facets to lay out the facets, adding some complexity to our pipeline in Fig. 9.1 on page 183, in which case the parameters on the statistics for faceting can be handled in the same fashion as in Sect. 9.2.4.

9.2.7 Modifying Transform Parameters

Transforms are a general data preparation tool and typically have parameters governing their behavior. We might bin or shingle data prior to faceting to allow us to do trellis displays [5], for example. Since the transforms are strongly data dependent (the data have not yet been standardized by the scales), interactive control of the parameters will depend strongly on the data and will need to be customized with specific data applications in mind. The general principles and applications of Sects. 9.2.4 and 9.2.5 will also apply in this situation.

9.3 Interacting via the Data

Time data are rarely simple. The point process and the univariate time series are the exceptions, not the rule. More commonly we have plenty of information, of varying types and maybe even at different levels of granularity. A simple time series of wind measurements becomes more complex when we try to understand relationships between:

- The geographical relationships between locations where the data were measured,
- Multiple time series for each location,
- Time series measured weekly at one location and daily at another.

If we have multiple time series, we want to see how one affects the other. In spotting patterns in a baseball game, we wonder if it makes a difference whether pitchers are left-handed or right-handed. When we look at long time series of stock values, we want to see if major world events have any affect on, say, oil prices.

When we see something interesting in views of time we have created, we want to explain it, usually by considering other data views or by including additional variables. Sometimes, it is not hard to add in variables and see if they can explain the feature, or indeed if they have any effect whatsoever.

[4]One potentially useful application would be if we transformed the categorical variable by aggregating all infrequent values into a single category, so, as an example for the email data, we might facet by data sender and have only the most frequent K senders retained. Other emails might be filtered out of the data or be given a single sender value like *other*. In this case, putting K under interactive control would be a useful exploratory technique.

A first thought might be to add a variable in as a positional variable – another coordinate for the element. If a histogram of X shows something of interest, you can "add" a variable Y to it by making a scatterplot of X against Y. If you want to explain something in a scatterplot, then it is possible to turn it into a rotating point cloud in three dimensions, and if we use parallel coordinates, we can proceed further.

Despite the undoubted utility of this approach, it does present some problems to the viewer. The main ones are as follows:

- As plots become increasingly complex, they become *harder to interpret*. Few people have problems interpreting 1-D plots. Scatterplots, tables, and grouped box plots or other displays involving two dimensions are easily learnable. But the necessity of spinning and navigating a 3-D point cloud or understanding the contributions to a parallel coordinates plot projection makes these views less intuitive than their simpler versions.
- It is harder to accommodate *differences in the basic types* of data. High-dimensional projection techniques assume the variables are ratio measure variables, as do techniques that display multivariate glyphs and, to a large extent, parallel axes techniques. Many high-dimensional views assume a uniformity of type of data, and most of them impose normalization on the data. With the variety of data that is present in time, and with normalization of data representing time at best an adventurous proposition, it can be hard to fit data into the desired "shape" for such a plot.
- Data that are of a type *specific to a particular domain* can be impossible to add directly. Exploring relationships in multivariate data collected at geographical locations, on nodes of a graph, or on parts of a text document is hard because of the difficulty of building views that correlate the statistical element and the structural element of the data. Often, two completely different procedures or even software tools are used for the analysis, with results from one package mangled to fit the input form of the other package – a task that is both frustrating and error prone.

If we have additional categorical variables, then we can use the faceting techniques of Chap. 5 to add those variables in, but this will make our plots much larger, and faceting is mainly useful for discovering conditional relationships – looking at how each plot is different conditional on the faceting variables. An alternative to adding positional variables was given in Chap. 7, where we gave methods for mapping variables to data using aesthetics.

For static plots this is about the best that can be done – there are only so many places in a chart into which we can throw a variable. A chart with several aesthetics, fully labeled with long informative names and faceted by several variables, is going to be hard to interpret, but with the printed page we're running out of options. In this chapter, however, we are not limited by such restrictions. Rather than adding features to an existing plot, an alternative is to use interactive techniques.

One method for using data interactively is to take variables and make them available for use as aesthetics that are not immediately used, but instead provide information for controllers and other interactive techniques to use. This is often termed *metadata*. A second technique is to augment the data with a variable that

represents the user's degree of interest in each row of the data table and then use that variable in the display. When the contents of that variable are under interactive control, and we use the same variable in multiple charts, this technique is known as linking.

9.3.1 Brushing and Linking

The basic idea behind *linked views* is simple; instead of creating one complex view, create several simpler views and tie them together so that when the user interacts with one view, the other views will update and show the results of such an interaction. This allows the user to use views that require less interpretation and views that are directly aimed at particular combinations of data. For time series data this is particularly useful since we often want to show our time-based variables in specific ways. For geospatial time series, the utility is even more pronounced, as we want both maps and time views to be available, and our interest is often in seeing how the two components interact.

One of the earliest linked-views work to achieve wide attention was the scatterplot matrix brushing technique of Becker et al. [6]. By arranging scatterplots of n variables in a table so that all the $n(n-1)$ ordered combinations of axes are present, the eye can quickly scan a row or column and see how a given variable depends on every other variable. This arrangement technique is enhanced by the use of a *brush*. A brush is a shape that is dragged over a visualization by the user and performs a selection operation on anything it moves over. The original implementation was to *paint* the data – when brushed, the data points brushed over are painted in a different color, both in the panel in which the brush is active and in all other panels of the window.

This technique embodies the basic principle behind linked views, the general methodology for which is as follows:

Define a pseudovariable: When data are being shared between multiple views, add a pseudovariable to the data that will represent the user's *degree of interest* in each row of the data. This variable will be controlled by the user; with the use of brushing, selection, or other interactive techniques, its contents will be modified.

Augment the basic chart using the pseudovariable: Using techniques described in other chapters, add this variable as a positional, aesthetic, or faceting variable for each chart that is to be linked.

Add user–interface gestures that modify the variable: Some way of allowing users to modify the variable must be designed. Brushing, clicking, hovering, using a menu, or editing data values directly are all candidates. When the degree-of-interest variable is modified, the linked chart should update.

The great strength of this technique is its versatility – all it fundamentally requires is a system flexible enough to be able to add a variable to a chart and then allow that

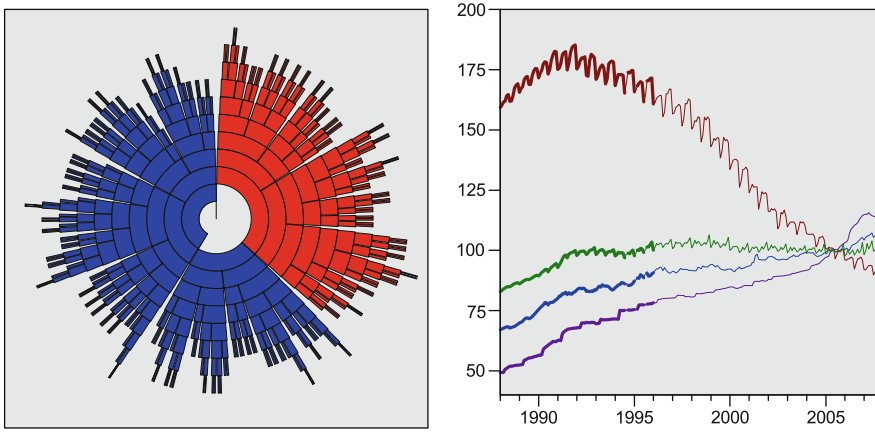


Fig. 9.10 A hierarchical clustering was built for the CPI variables food, clothing, housing, and furniture and is shown using a radial space-filling layout (*left*). The raw data for these variables have been plotted against time (*right*). A section of the clustering has been selected and shows in *red* (*left*) and as *thick lines* (*right*). It would be preferable to use the same aesthetic to display the selection in both views, but the difficulties of using the same aesthetic for lines and polygon elements forces compromises to be made

variable to be put under interactive control. It is even *almost* possible to link two charts in Excel.

The linked-views paradigm is both general and powerful. It has been explored in more detail in papers such as [132] and [140] and has been used to link specialized displays of specific types of data, as was done in [50] for geographic brushing. The rest of this section will show an example of linking domain-specific charts: tree displays for hierarchical clustering linked to time series. The modeling techniques of earlier sections will also be incorporated. At the end of the section will be a couple of short subsections describing extensions to this basic use of linked views, but these provide only a sample of the techniques that can be used.

The most common, and arguably the best, user interface for linked views is to allow users to brush the data as shown on the chart or lasso sections of it – a direct manipulation of the chart. The simplest result of such an interaction is to select all the items under the brush or in the lasso and to make the others unselected. The pseudovariable is a simple boolean, and any chart that is linked must distinguish data based on that flag. This selection interface is used in the following example.

Figure 9.10 continues with the example of CPIs from the UK. A hierarchical clustering has been performed on the months, based on all the CPIs in the data set except the overall CPI. On the right is a multiple time series chart showing four indices (food, clothing, housing, furniture). Both views are linked to each other so that selecting a region of one chart highlights the corresponding months in all other views.

The hierarchical display shown on the left is an example of a *domain-specific* display. It is a visualization that is tailored for one form of data only – a display of a

tree where each node of the tree has a size, and those sizes are additive. The display mechanism is a radial *space-filling layout*. Each level in the hierarchy is represented by a band at a set radius, with the root node in the center and children outside their parents. The angle subtended by a node is proportional to the percentage of the entire population that this node represents. Children are laid out directly outside their parents, so parents "divide up" the space for their children according to the children's sizes.

In this case the tree is a *hierarchical clustering* tree – the leaves of the tree are the original data points, and these are successively grouped into clusters based on how similar the data items and clusters of data items are. If you look at a band near the center, the 360° arc is divided into a few groups. Figure 9.11 gives an example of this type of display in a different context and should help clarify the layout technique. Graph layout is a complex discipline, and a friendly introduction can be found in [4] with an extensive annotated bibliography available in [33]. Some useful chapters can be found in [18] that use the same algorithms as shown in this chapter.

In Fig. 9.10 a cluster has been selected by clicking on it; the cluster chosen was a central cluster containing about one-third of the data points (the arc has an angle of about 120°). All the data points that are contained in this cluster, and hence all the smaller clusters that make up this large cluster, are drawn in red. This selection is linked to the time series display, where the selected cases are shown in a thicker style (using the size aesthetic). The result is immediate and obvious – the linked cases all occur in the same time region at the beginning of the data set.[5] This cluster, found by considering just CPI values and not time, is perfectly defined by the time variable. This example shows the power of the linking paradigm. One view shows a highly specific view of the clustering with no time information; the other view shows a classic time series view. Linking allows us to explore and make inferences about the relationship between these two aspects of the data without major modifications to the views themselves.

One hypothesis might be that the clustering is completely defined by time; maybe all the clusters just represent slices of time. Easy enough to check – just select a new cluster. We choose a new selection for the linked views, selecting a cluster one step further out. It and its children are all shown in red. The result is shown in Fig. 9.12. The *parent* of this cluster is shown in green, since only half of its items are selected (green is halfway between red and blue on the scale of this implementation of the color aesthetic). The linking shows that although there is a slice of time that is strongly related to this cluster, the relationship is not perfect; the selected months do not form a contiguous set.

In Fig. 9.13 a third chart has been added to the existing two. At the top right is a histogram showing residuals from a fitted time series mode of the overall CPI. Each view is linked to every other view, so that selecting a region of one chart

[5]Since this chart shows time series for each variable by each month, a single row of data is shown in all four lines, which is why the selection is duplicated in each line.

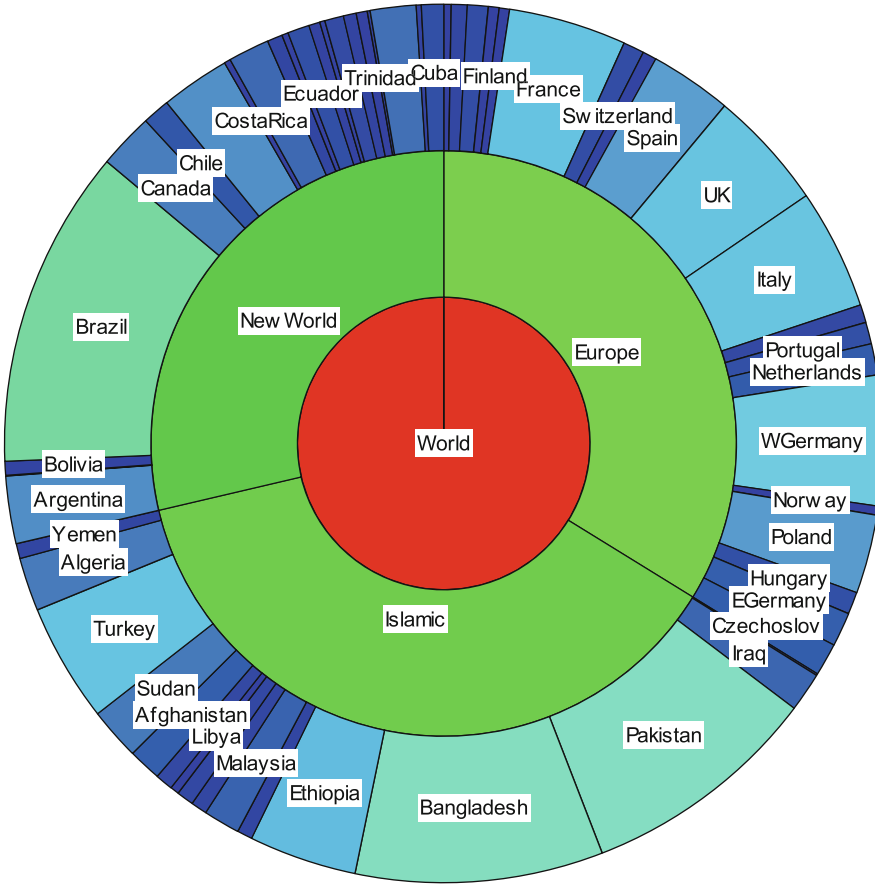


Fig. 9.11 A subset of world countries, laid out in a hierarchy and grouped by their region variable. In contrast to the hierarchy of Fig. 9.10, the hierarchy here is defined a priori by the data and not imposed on the data. The basic layout is, however, the same. The *leaf nodes* on the *outside* (the countries) are placed outside the group in which they are clustered (*regions*). These are then clustered into a single *root node* – the world. We could put intermediate levels in the hierarchy or added zip codes for some countries, in which case we would have had a less regular display than this simple example. The color of the *regions* represents population size and so presents the same information as the angle

highlights the corresponding months in all other views. The model is similar to the one we derived in Sect. 9.2.4.1 on page 192, except for the overall CPI, not for the furniture CPI.

In this figure, the higher residuals have been selected, denoting months where the actual CPI was higher than the model predicted. This is shown in red in the histogram, and in the time series view the segments corresponding to those months are shown with a thicker stroke. It appears that these residuals occur mainly on the downward part of regular cycles on the topmost time series (representing the

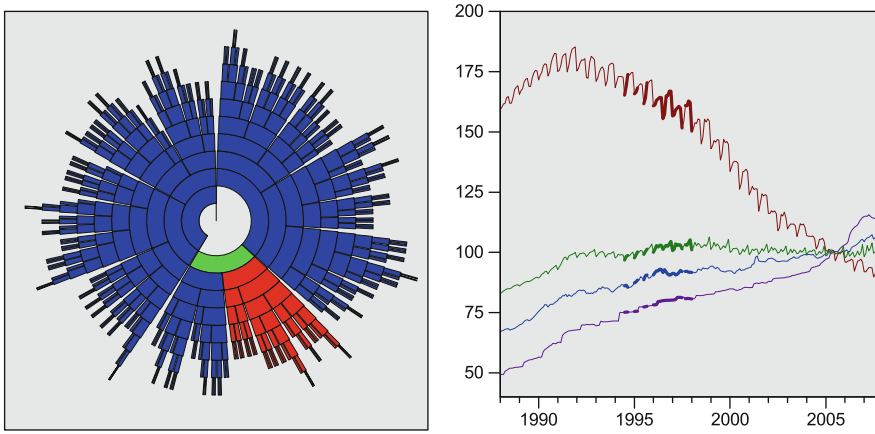


Fig. 9.12 The same figure as Fig. 9.10 on page 199, except with a different selection. Here a smaller cluster from further down the hierarchy has been selected. This results in the parent of the cluster being semiselected, and so gets a color midway between *red* and *blue* on our rainbow color scale – *green*, in this color aesthetic system

clothing CPI). In the hierarchical view, partially selected clusters are drawn in color hues between red and blue. Because the months are aggregated at higher levels of the hierarchy, the interior nodes in the hierarchical tree map may be partially selected. As before, the selection percentage is shown with a rainbow hue map, with blue indicating completely unselected, red completely selected, and green half-selected. Intermediate hues indicate intermediate selection percentages. At the outer levels some fully selected clusters can be seen, but more interestingly, looking at the inner bands, there are strong differences in color, indicating some relationship between the distributions of the residuals for the overall CPI, and the distribution of clusters of the other CPIs. This indicates that some combinations of the individual CPIs have a different effect on the overall CPI than other combinations. Looking at the linked time series, there is some evidence that the model under-predicts CPI when the individual CPIs are on the decreasing section of the regular cycles.

9.3.1.1 Selection Details

In the implementation suggested above, selecting a set of data points makes them the subset of interest and defines the rest of the data as of no interest. This makes it quite hard to refine a selection if you miss points, or want to add some points, or even just make a couple of disjoint selections. It is therefore almost necessary to allow the user to modify selections. Given an existing selection, the user should be allowed to add cases to or remove them from the selection. This is essentially a boolean operation – for each case, it is either selected or unselected initially, and

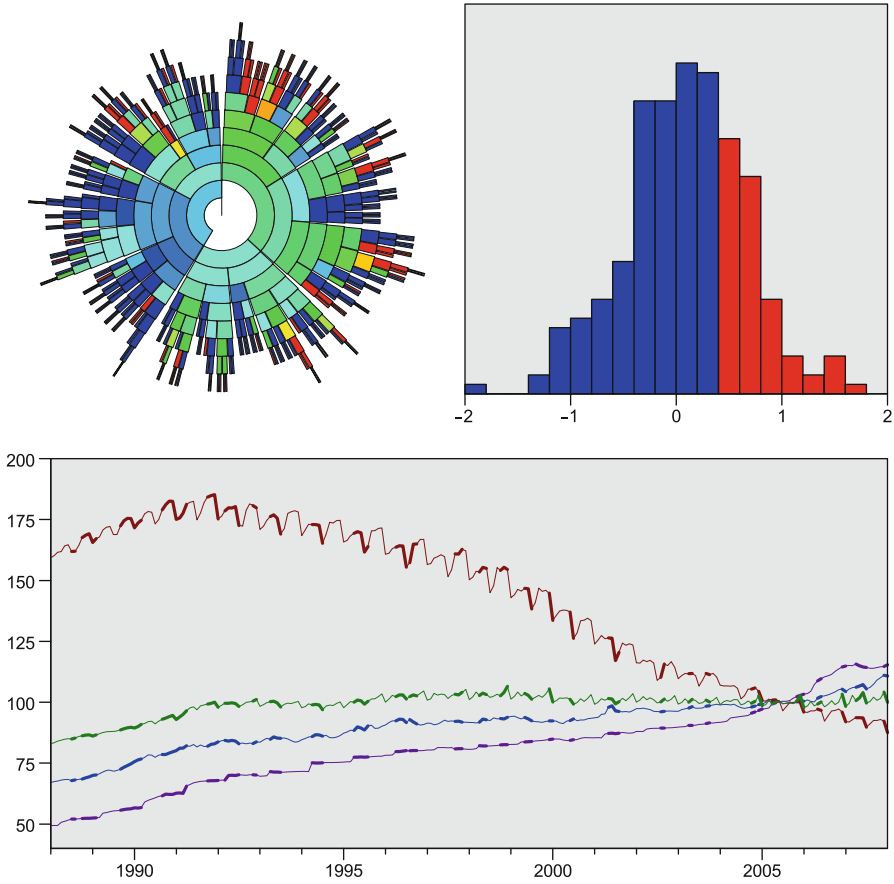


Fig. 9.13 Hierarchical clustering of CPI data. This figure shows the two views of Figs. 9.10 on page 199 and 9.12 on the preceding page rearranged, with a histogram in the *top right*. This histogram shows residuals from a fitted time series model for the overall CPI. The higher residuals have been selected, showing in the other views months where the observed CPI was higher than the predicted CPI

then the brush will either contain it or not contain it. These two boolean states are to be combined to produce a selected/unselected outcome. This should be a familiar situation to any computer scientist readers – it is simply a boolean operator.

The most common operation is replace, which simply ignores the previous selection. Various boolean operators correspond to other common operations: intersect (AND), union (OR), toggle (XOR), complement (NOR), and subtract (NAND). Wills [140] investigates which combinations are necessary to provide a useful system, concluding that the following are most useful, in order of complexity:

- *Replace*
- *Replace* and *Toggle*

- *Add* and *Subtract*
- *Add* and *Intersect*
- *Replace*, *Add*, and *Subtract*

9.3.1.2 Nonboolean pseudovariable

It is not necessary to force selection to be a boolean state. The notion of selection has been extended [132] to be continuous so that cases have a selection status lying between zero (not at all selected) and one (completely selected) and implemented in tools such as Xmdv [76]. For time series, we might define a time point at which we have a strong interest, and then use a "window" approach to say that data points with a time outside the window have no interest, whereas data points inside have a higher degree of interest if they are closer to the critical time. The choice of window size and kernel function with which to assign a degree of interest has clear parallels with density estimation. Given how often an interactive graphics technique can resolve to setting an interactive window width and applying it to parameter selection, histogram bin width, or general density smoothing, brush sizes (for time series and geographical[6] systems among others) and dynamic filtering, it might well be claimed that this is the single most powerful interactive technique.

9.3.2 Drill-down

Drill-down is a technique to move from a summary or aggregated view to a detailed view by focusing on part of the original view. It is a *details-on-demand* technique, but it differs from a *focus+context* technique in that the focus (or detail) view is not combined with the context but instead is shown as a separate view. Drill-down can be considered as a special form of linking; the second view is linked to the first by filtering the data; the selection in the summary view defines what data will be used in the secondary view. The strongest difference between a linked view and a drill-down view is that in a linked view the data that are of interest (the selected subset) are shown in relationship to all the data; they are simply identified differently. In a linked view, even if visibility were used to show the selected subset, the plot area would encompass the whole range of the data, whereas in a drill-down view, the details view shows only the data that have been selected; the rest of the data is of no relevance, and so scales for both positional variables and aesthetics default to show only the data in the selected subset. Figure 9.14 shows this comparison between *drill-down* and *linking*.

[6]For spatial data the window width can be used to define a geographic region, typically a disk, that can be moved over a map to highlight data that lie in that region. An early example of such an interactive geographic exploration system is given in [50].

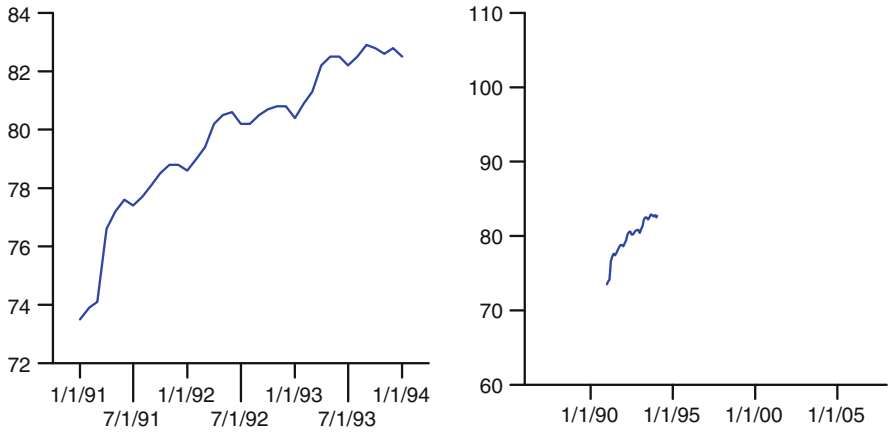


Fig. 9.14 Drill-down compared to linking. Drill-down into the range $1991 \leq year \leq 1993$ (*left*) compared to a *linked view* with the same selection (*right*). The drill-down "zooms in" to the selected data, showing details more clearly at the cost of losing context, whereas the linked view, using the visibility aesthetic, retains the original scales and simply shows the selected data differently. In this case "differently" means "at all"!

The choice of which to use is decided by the goal. If the summary view provides a good overview of the context, or that information is not expected to be of interest once a selection has been made, then drill-down's tight focus is valuable. However, if the context is still of interest, or if sudden changes to the focus view are undesirable (for example, when animating over a selection in the summary view), then the linked view is preferable.

9.3.3 Summary

The number of possible interactions that can be added to charts is effectively unlimited. In this chapter we have shown a selection of different interactions, from simple examples such as adding pop-ups to complex linked displays like Fig. 9.13 on page 203, which links residuals from a time series model to a hierarchical clustering view and a multiple time series view. These examples have been placed in a framework that considers interactions as either modifying parameters or adding metadata.

The framework for describing a visualization as described in Chap. 2 was presented as a pipeline, with the classes of interaction listed above considered as orthogonal attributes that can be used to modify each framework component. The result is a framework for interactivity that is both *descriptive* – an interaction can be described by whether it affects data or parameters and by which framework component it affects – and *prescriptive* – given a desire to interact with a given chart feature, the interaction framework gives the designer the information needed

to implement this feature. As a bonus, the position in the display pipeline provides information on how efficient the interaction will be. If it is late in the pipeline, interactions can be fast and require little re-execution. If it is early, more re-execution will be required. The following table lists the specific interaction techniques described in this chapter, indicating their type and which component of the pipeline they affect.

Interaction	*Type*	*Component*
Magnifying glass	Parameter	Element
Color mapping slider	Parameter	Aesthetic
Pop-up	Parameter/data	Aesthetic
3-D rotation	Parameter	Coordinate
Spherical map projection	Parameter	Coordinate
Distortion/fisheye	Parameter	Coordinate
Histogram bin widths	Parameter	Statistic
Seasonality fitting	Parameter	Statistic
Choosing transforms	Parameter	Scale
Brushing scatter matrices	Data	Aesthetic
Linked views	Data	Aesthetic/position/faceting
Drill-down	Data	Data

As with any realistic classification, there will be gray areas where an interactive feature might not be easily classifiable (such as we noted for pop-ups), but the same is true for any language, as can be seen by considering the quotation that heads Chap. 5. As always, the goal is not to dictate conformance but to provide insight and drive good visual presentation of time data.

9.4 Further Exploration

One of the most fertile sources of information on interactive graphical techniques are the *Proceedings of the IEEE Symposium on Information Visualization*. This conference has been going since 1995 and contains a taste of most visual techniques. A lot of the papers tend to be of the "show and tell" school, especially those from the earlier years, but the trend is positive and many good papers have been published in the proceedings.

There are a number of worthwhile books on interactive graphics, which seems more often termed "dynamic graphics" in book titles. A recent example is [143], and [23] is a classic, but it would be an unusual book on visualization published since 2000 that didn't mention interactivity.

Graph layouts are a visualization technique that can be very powerful, but it appears underused. With modern computing power they are easier to generate for moderate sizes of data than ever before. Di Battista's *Graph Drawing: Algorithms for the Visualization of Graphs* [4] is an accessible introduction and should pave the way for more thorough investigation.

Chapter 10
Topics In Time

Time, with his innumerable horse-power, worked away, not minding what anybody said, and presently turned out young Thomas a foot taller than when his father had last taken particular notice of him.
"Thomas is becoming," said Mr. Gradgrind, "almost a young man."

Time passed Thomas on in the mill, while his father was thinking about it, and there he stood in a long-tailed coat and a stiff shirt-collar.
"Really," said Mr. Gradgrind, "the period has arrived when Thomas ought to go to Bounderby."

— Charles Dickens, *Hard Times (1854)*

10.1 Large Data Sets

The definition of a "large data set" is not a fixed one. At one end, data-intensive statistical or mining techniques might start to run out of steam with a few tens of thousands of values. At the other end, a database expert might consider simple tabulations of data on a billion cases as only getting toward large. And if you are reading this book a decade or so after it has been published, you might well be glancing at an analysis you are running on a public cloud computing platform and be mildly amused at these statements.[1]

For this book we will take what I would consider a "moderately large" data set as an example. The data were published at the 2009 Statistical Computing/Statistical

[1]Being a fan of Golden Age science fiction makes it painfully clear that any guesses about future technology might lead you to be the subject of much hilarity in the future, as in E. E. "Doc" Smith's Lensman series, where the title characters rocket around the universe at enormous speeds, plotting their courses with the aid of extremely precise slide rules [101].

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_10,

Graphics Data Expo [1] and are available at the time of writing for public download. The data consist of flight arrival and departure details for all commercial flights within the USA, from October 1987 to April 2008. There are almost 120,000,000 records, and the files take about 12 gigabytes of data uncompressed. This size data can comfortably fit in any reasonable database, and we could carry it around on a fingernail-sized memory chip if we needed to, but for visualization purposes it can be considered large under one useful definition: We never want to plot the data except in *aggregate* or when *filtered*.

The goal of this study was understanding, a desire to learn more about the data. In particular, the goal was to understand what causes flight delays. The American Statistical Association site for this expo [1] contains a number of example analyses, all of which are heavily dependent on aggregation and filtering to facilitate visualization.

10.1.1 Aggregration

Figure 10.1 gives one view of the data, but this was by no means the first chart created in the analysis. The steps needed to create this visualization included:

- Creating univariate views of each field and deciding suitable transformations for each;
- Building database views that rolled-up the fields in different ways;
- Tailoring the view to the data.

Decisions made on how to encode or recode data can have a large effect on the analysis. For this data set there were multiple measures of delay (when in the flight process they occurred – in-flight, at the departure airport, or at the arrival airport) as well as codes to indicate a flight as diverted or canceled. In this chapter we have recoded these variables into a single measure of the amount of "pain" caused by delay. There are seven levels of delay, using an exponential scale (with breaks at 0, 15 minutes, 30 minutes, 1 hour, 2 hours, 4 hours) and, in increasing order of pain, two categories for diverted and canceled flights. This gives us an ordinal scale of delay, where the lower levels mean relatively little nuisance and the highest level most nuisance.

Three-dimensional views are often a poor choice, especially when there are more than two or three categories on each axis, but in this case they work well due to a fortunate, but somewhat expected, characteristic of the data: The worse the delay, the fewer there are of them. Canceled flights form an exception to the rule (for this study they were coded as the worst possible flight result), but overall the figure works fairly well. We can see from it that, in general, the distribution of the severity of the delay does not appear to be related to the month – either months are good for delays in general or they are bad. If we displayed each month as a bar or line

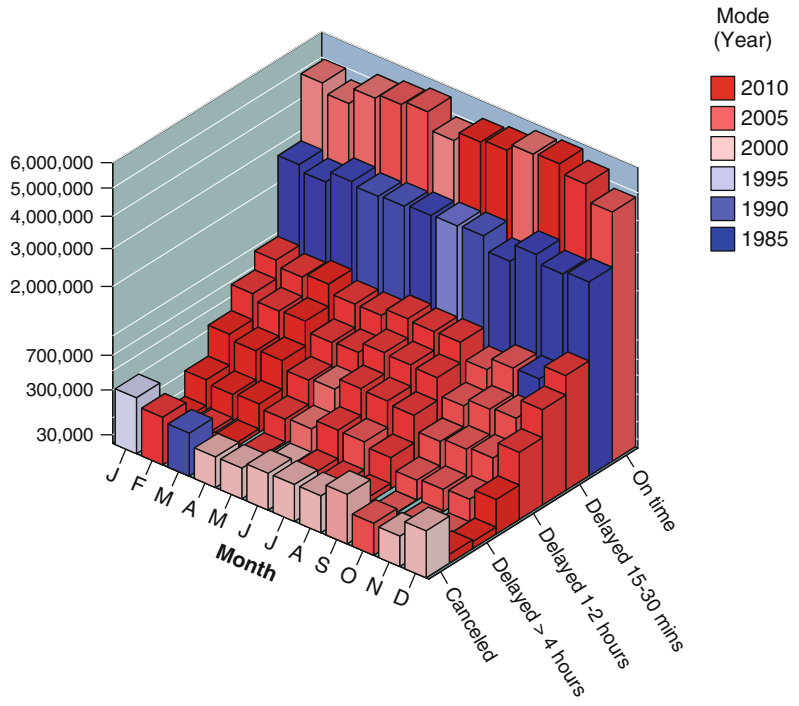


Fig. 10.1 Delay counts by severity and month of year. The data for this figure were rolled up in the database. SQL versions of the date transformations given in Table 8.1 were used to calculate the month in which each flight took place, and the delay duration was categorized into one of nine different ranges. For each resulting table cell, the most commonly occurring year was also recorded and used to color the 3-D bar chart. Note also the nonlinear scale on the *vertical axis*, allowing us to see features in the less-frequent delay classifications. The year has been encoded as color on a divergent scale, with early years in *blue*, later years in *red*, and years in the middle in *white*

chart and overlaid or paneled the 12 charts as we do in Sect. 6.4.1, we would see this more clearly. There is also not much interesting going on with the years – with two exceptions.

The first exception is in the reporting of very short delays – which seems to occur more in the eighties than other types of delay, as shown by the uniform blue band running along the category of delay for 1 to 15 minutes. That might be explained by different reporting practices (we will see another example of this later). The other interesting point is the fact that the worst year for cancellations in March was early on. Mousing over the bar, which has a tool tip interaction to give full details, lets us know that the year was 1989. Figure 10.2 was built to explore cancellations in more detail.

Figure 10.2 drops the information on types of delay and concentrates on (filters) canceled flights. The initial version showed a well-known exceptional value, so the

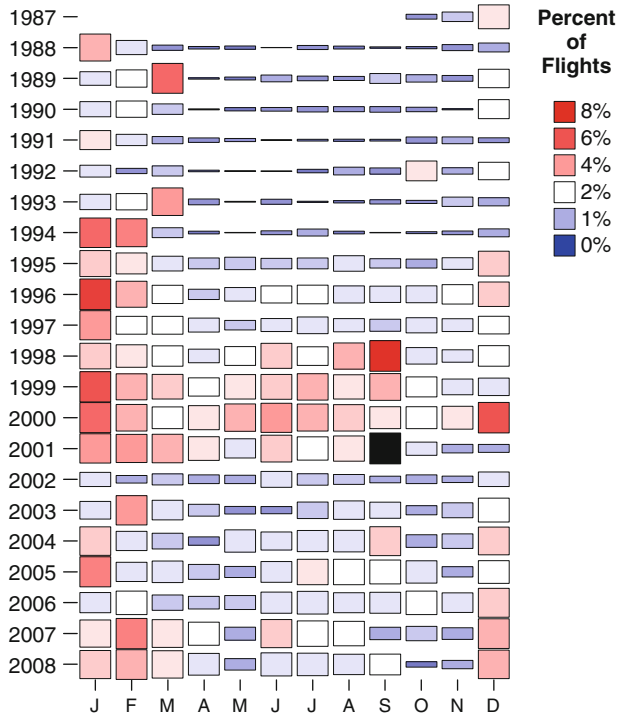


Fig. 10.2 Canceled flights by month and year. This chart is tabular in format and uses two different time measures, month and year, as the two dimensions of the table and then displaying an interval at the cell location. The height of the *bar* encodes the count of canceled flights in that month/year, and the color encodes the percentage of flights that were canceled. September 2001 is an extreme outlier for percentage of canceled flights, so the domain for the color aesthetic has been reduced to exclude that value, which is shown in *black* to signify it is out of domain

display was adjusted to filter that information out of the color mapping.[2] The result is an informative chart that shows a number of features:

- The unusual number of canceled flights in March 1989 is confirmed. Usually March is not too bad for travel, with only 1993 showing problems. For a possible explanation of the data for this month, search online for "March 1989 Quebec Blackout."

[2]Another detail to be considered is the color mapping. The scheme used in this figure is a blue-white-red divergent scale, but an alternative would be to use quantiles of counts (such as banding the data into the lower 20%, the next 20%, etc.) This would direct attention to the overall distribution, whereas the mapping shown here highlights extreme values – in this case, those flights with a high cancellation percentage. Which one to use depends on what part of the distribution you are concerned with; quantiling highlights the middle, a simple scale highlights extremes.

- Up to about 1995, flight cancellations occurred mainly in winter, but from 1995 onward it appears that cancellations started increasing in all months (although a likely explanation is that 1995 is when uniform reporting of flight delays was first introduced). After September 2001, the situation changed dramatically, and it took until 2007/2008 until the data began to look like they did a decade previously.
- May, October, and November are the best flying months. No wintry weather and no summer congestion make for a more likely flight completion.

September 2001 was the worst month for flying. The second worst month was also a September – this one in 1998. There are a couple of possible explanations for this. One is that that month was a bad one for hurricanes – at the end of September, for the first time in recorded history, four Atlantic hurricanes formed simultaneously: Georges, Ivan, Jeanne, and Karl. Two of the hurricanes in that month had their names retired since they were so devastating. However, even severe hurricanes do not usually affect cancellations to the degree shown here. A more likely explanation is that Northwest Airlines, a major carrier, had a long pilot strike that month (as did Air Canada, which may have also impacted flights to a lesser extent). Between these two effects, it was not a happy month for air travel.

The format of Fig. 10.2 is a useful and flexible one. With the many ways that time can be broken down – century, decade, year, quarter, month, week, day, hour, and so on – it is not hard to make a tabular format that places an aggregation of time on one dimension and the next finest division of time on the other dimension. If a suitable level of aggregation is chosen, then a table can be made that has a reasonable number of divisions on each dimension with the divisions being interpretable.

Figure 10.3 shows a reworking of this basic format at a different level of aggregation. Instead of year by month, the dimensions of the chart are month by day, but otherwise the visualization is the same. This illustrates an important point; when analyzing large data sets, it is very common to make many forms of aggregation and display many different visualizations for these aggregations. Rather than fall prey to the temptation to make each one as individually good as it can be, it is more helpful to the viewer to keep the same basic building blocks in each one; consistency is the better goal.

Figure 10.3 differs in one subtle way from the previous version (apart from the location and orientation of the legend). Take a moment to study the two before reading the next sentence. The difference between the two is that in the previous version the width of elements is less than the available space, so they do not touch. In Fig. 10.3 the bars do touch. This makes the horizontal dimension much stronger – the bars visually merge together horizontally to make the chart seem more like a set of complex glyphs that show months, with those glyphs varying internally. In this figure it is hard to compare days to each other; our visual processing system makes this figure appear as 12 horizontal items, rather than the more separable rectangles of the previous figure. A minor detail in the chart makes a major perceptual difference, aiding our ability to make one form of comparison and hindering our ability to make another one.

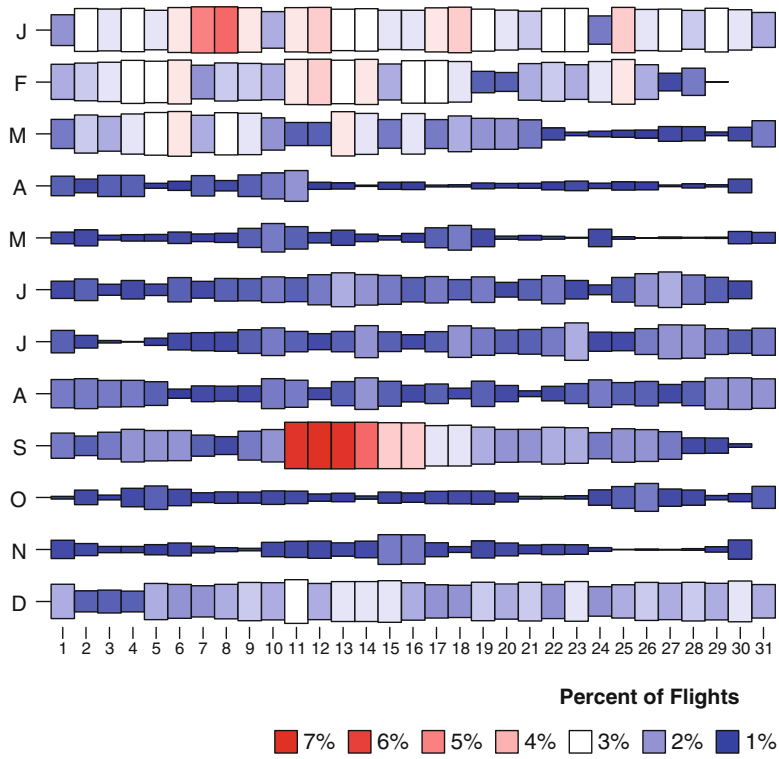


Fig. 10.3 Canceled flights by month and day of month. This chart is tabular in format, using two different time measures, month and day of month, as the two dimensions of the table and then displaying an interval at the cell location. The height of the *bar* encodes the count of canceled flights in that month/year, and the color encodes the percentage of flights canceled

Figure 10.3 is the figure from my study of this data set that I think is the most useful. I rarely get a chance to decide which year to travel in, but I often have a fair amount of flexibility in the day or even month of travel. This chart gives some helpful advice:

- *Winters are bad for travel*. December through February show high rates of cancellation (except for February 19 and 20, of which more below). March, as the saying has it, "comes in like a lion and goes out like a lamb."
- *Summers are slightly bad*. A comparable chart for delays, rather than cancellations, shows that summer suffers from delays more than cancellations. You may be stuck in a hot airport for hours, but you have a better chance of getting out than if you'd traveled in winter.
- *September 11*. The one outlier year of 2001 causes this chart to be dominated by that effect. By canceling every flight for several days that year, it effectively boosted the percentage of canceled flights for that day (and the following days)

by 5% as measured over the entire time period. To minimize this effect, we might not plot the overall percentages, but instead plot the median of the 20-odd cancellation percentages for that day.
- *Holiday travel is good*. Because December 24/25 and January 1 are in deep winter, travel then is only slightly better than surrounding days, but traveling on July 4 is excellent, and even Washington's birthday (the third Monday in February – often around February 19) shows decreased cancellations. Memorial Day weekend (the last Monday in May) also shows decreased cancellation rates.

Overall, the best months to travel are April, May, October, and November – between the bad weather of winter and the summer vacation travel. Traveling on holidays is also a good option, if you can manage it.

10.1.2 Augmenting Traditional Displays

The table is a traditional tool that has been used in summarization of large data for thousands of years. We have used it as motivation in Figs. 10.2 and 10.3, but if you need to output simple noncolor summarizations, we can drop the size aesthetic and modify the color aesthetic to produce grayscale and produce a visualization such as Fig. 10.4.

One feature that is included in the chart was that of outlining "significantly large" cells. Using measures of significance to focus attention on subsets of data seems a good idea initially, and it certainly has a long history. However, significance values are tricky things to use to make decisions. When using a p-value, it is important to be aware of its failings as a diagnostic tool.

- Vast numbers of people are unable to understand what a p-value actually represents and will make serious errors in their interpretation (see [95] for a fairly technical overview, or check Wikipedia for a rough idea of common errors).
- There is a difference between significance and importance. The airline data set has 120 million records – pretty much any effect will be significant to several decimal places. But when you are planning a trip, do you care about an effect that is 99.9999% likely to be real but makes an expected difference of less than 3 minutes to less than 0.01% of flights?
- The use of p-values at fixed levels leads to seemingly paradoxical statements. It is not hard to find examples where group A is significantly different from group B, and group B is significantly different from group C, but groups A and C are not significantly different. In Fig. 10.4, there is no significant difference between February 11 and 12, and yet one is significant and the other is not.

In the end, the ubiquity of p-values and their history of usage within tables led me to leave them in the figure. Also, because we are showing the results of many such tests, the true significance is hard to evaluate, and really all we are doing is

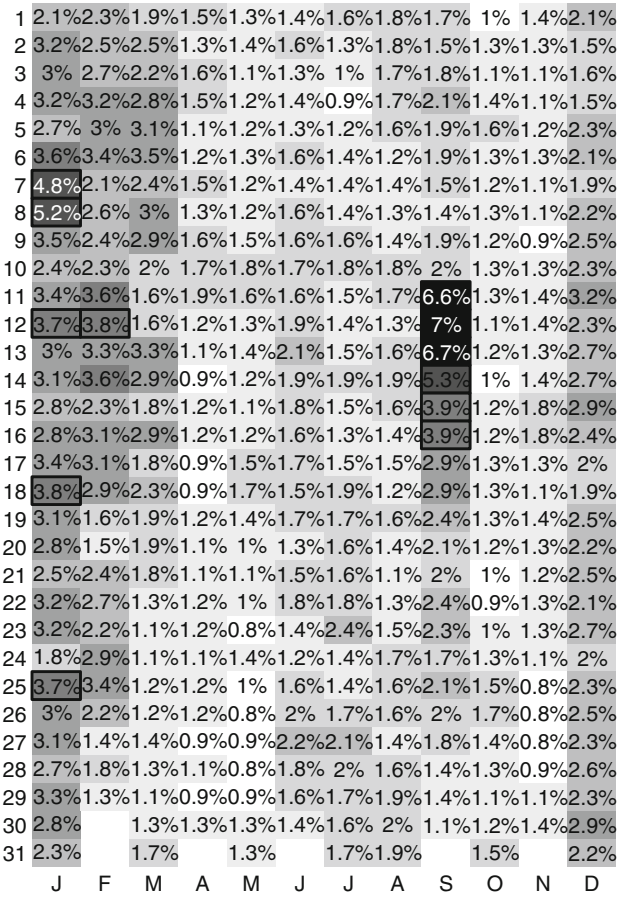


Fig. 10.4 Canceled flights by month and day of month. Percentage of flights canceled by day of year. The *background shading* redundantly encodes the label data so that human visual systems can do their work more easily. *Cells* that are significant at the 5% level are outlined with a *thick border*

highlighting those values higher than some arbitrary fixed mark – the significance of a single cell, given the repeated measures, is not truly likely to be as stated, nor does the base distribution fulfill the required conditions – all we really have is a cutoff that is motivated by a statistical concept. If it helps us see patterns, we can consider it a useful tool.

Figure 10.5 is another traditional tool – the basic time series chart. We have paneled it by year and applied a filter, but otherwise it only differs from the traditional time series chart in one respect: Each line segment has been colored by the difference between the traffic in that month and the traffic in the same month of the previous year. This allows us to discount seasonal effects and compare

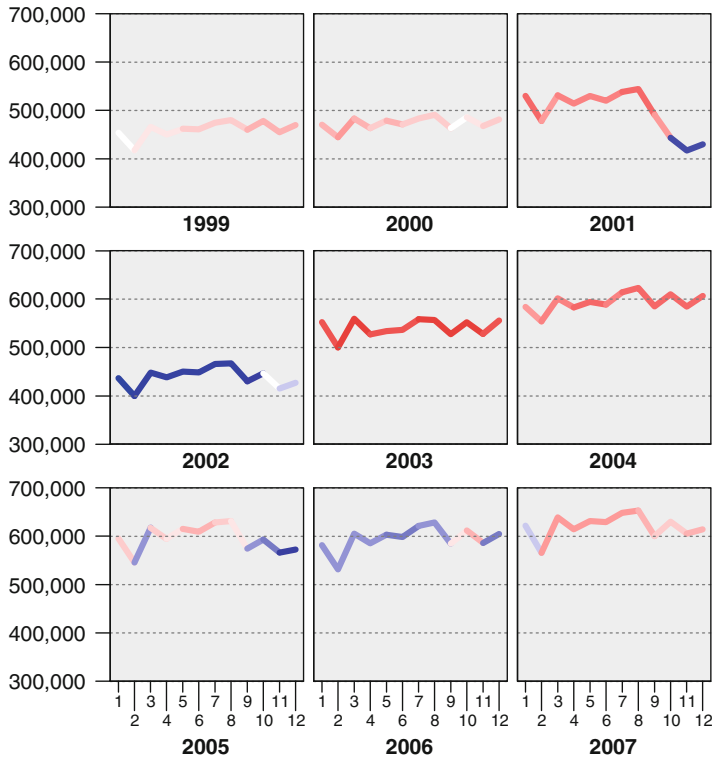


Fig. 10.5 This figure shows the total number of flights on a monthly basis. An interactive filter was applied to the data allowing the analyst to show panels only for a given range of years. In this snapshot, the years 1999 through 2007 have been chosen. The color of the *line* indicates the relative change from last year's value, with *gray* meaning no change, *blue* a decrease, and *red* an increase in flight frequency. This chart also serves as a quality check on our data; if we compare the data shown on this chart to other sources (for example, the Bureau of Transport Statistics), we see that this data set does not contain the same set of flights; the BTS data indicate there were significantly more flights, so we should treat this figure with caution

year-to-year changes more easily. Again, this is an example of taking a well-known chart and enhancing it for a specific task. Simple filtering and an added aesthetic make it much more useful than the basic chart would be.

Another traditional display tool is the map. In Fig. 10.6 we show a map of the USA with simple overall summaries of the total numbers of flights into and out of that airport over the study. Just as all large data sets have a time component, most of them have a spatial component. All data are collected at some time and some place, so it is not uncommon for geography and time to be found together. One study states that nearly 80% of data stored in corporate databases has a geographical component [60].

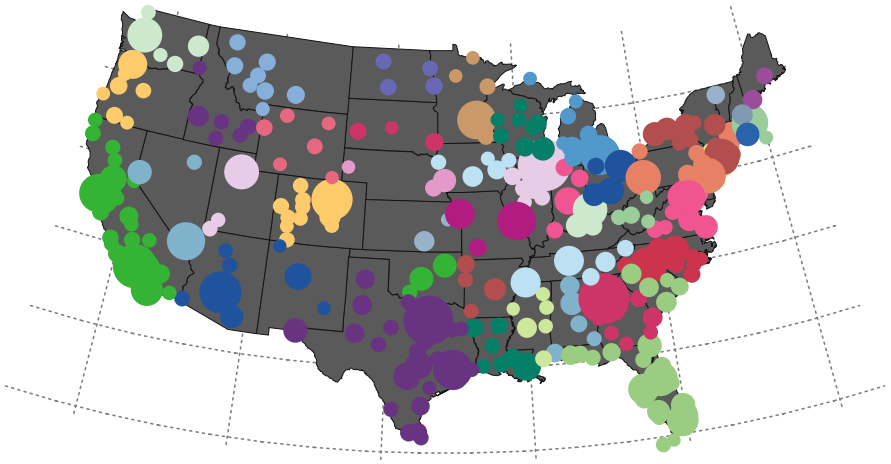


Fig. 10.6 Busy airports. Locations of commercial airports, with the sizes of *circles* (their areas) proportional to the total number of flights originating from them over the entire time period. The airports are colored by the state in which they are located, which helps a little in distinguishing airports in busy locations

In a full system, we might want to show both Figs. 10.5 and 10.6 together, allowing selection on one to affect the other – linking the two together via the general paradigm described in [32, 138, 141].

In general, visualization systems that combine spatial and temporal data explicitly benefit strongly by having an interactive component. An early and influential system was *SeeNet* [7], shown in Fig. 10.7, which explores telephone call traffic in the USA – a similar data set to the airline traffic data set being discussed in this section.[3]

This figure has a time view, which is used to filter the many links to show only those for a particular time step. There is also a filter that restricts the view to links that have a defined range of overload. There are also a number of controls that allow the user to change the display to compensate for undesirable display features, such as the tendency for long lines to dominate the display. It depends on interaction to achieve its goal of understanding how telephone calls overload a system immediately after a significant event.

This display is not a traditional one, but it is built on traditional components: a map, a graph layout, a time series view, and some legends. When designing a system for exploring a large data set, it is not necessary immediately to rush to a new and radical display technique. Building on existing knowledge establishes a base against which new techniques can be tried and evaluated. "Classics" are classics for a reason – they have stood the test of time and they work.

[3]This paper also provides an example of the way the term "large" is used. The data are described as challenging due to the large number of links – in 1989 over 12,000 links was considered a challenge even for serious computers, as were time data recorded every 5 minutes for a day.

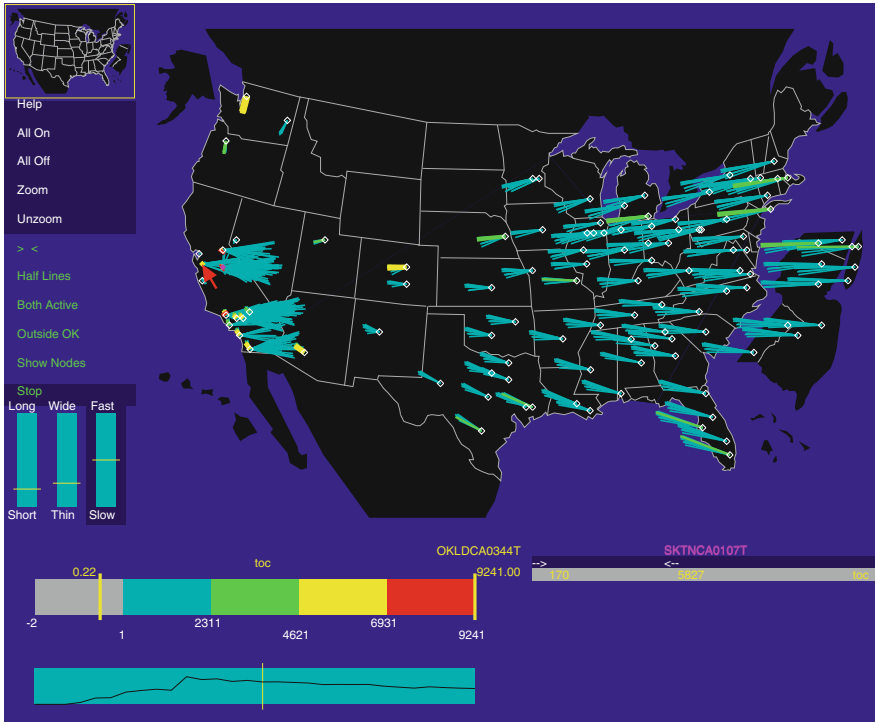


Fig. 10.7 Spatial/temporal telecommunications traffic: network traffic data for telecommunications traffic among the 110 switches in the AT&T network on October 17, 1989, the day of a major San Francisco earthquake. The data have both a spatial and temporal component, with traffic levels recorded for every 5 minutes. The map view shows telephone call flow, color coded by the amount of "overload" traffic amount – in this case mostly east–west, with some intra-Bay Area overload

10.2 Time Lines and Linked Events

The default goal when studying data is to understand them better – to see patterns, spot unusual values, identify trends. Time is typically seen as a predictor of something else, as we saw in the previous section where we wanted to know what the effect of time was on flight delays. It is more unusual to have time as the target of a visualization – asking what determines a given time. Durations and time extents are much more common; it would be very reasonable to ask, for example, what affects the duration of a flight. For time extents, we are not using much of the temporal nature of the data. Time is being treated as a simple scalar value, a magnitude quantity no different from width, height, price, or any other simple measure.

For unrelated events, visualization of time extents is therefore not significantly different from standard visualization. The situation becomes more interesting, however, when we consider a set of related time events, each of which has duration. Sometimes the relationship is defined by the problem data themselves. A Gantt chart

is an example of a chart that shows such a relationship. It displays the status of a project, where the project consists of various tasks, each of which is expected to take a certain amount of time, and where tasks have a dependency so that, for example, a given task cannot start until some other tasks have been completed.[4]

10.2.1 Linked Events

Figure 10.8, although a somewhat whimsical example,[5] shows how this type of chart lends itself less to understanding the data and more to understanding the process by which the data are created. The result is suited less for an exploratory goal and more for a goal of planning activity and allocating resources. This figure uses data on a tabletop role-playing game, where players take on personas and achieve sets of goals, with achievements in one game session carrying forward to the following ones. Each session, a player may play one module (the base unit of an adventure, often termed a "quest" in online gaming) with one of their personas (also called "characters"). Certain levels of achievement are necessary for a player's persona to qualify to play a given module. Some modules are standalone, some are linked into a series. From a player's point of view, it is desirable to have one's character play complete series when possible and to know what modules are available for a given level character. Players may also be interested in what virtual part of the game world the module takes place.

From the point of view of the people creating the modules (the company producing the adventures and distributing them to their customers), they have a fixed set of resources available. The writers are divided into teams, one for each virtual world area (coded as the game type on the y dimension in Fig. 10.8, and they must manage the process so that as players' personas achieve more and their "level" increases, they have modules they can play (and thus remain invested in the game and purchase more of the companies' product). Looking at this chart, which shows the start of the project, known as "Living Forgotten Realms," we can see that most writing groups have a similar load but that some prefer linked modules, whereas others prefer standalone ones. We can see, via the waves of color, that the designers expected that personas would gain levels at the rate of about one level per month, and that they expected that players would start playing more second personas

[4]This type of chart seems to have been invented first by Karol Adamiecki, who called it a *harmonogram*, but he didn't publish it until the 1930s, by which time Henry Gantt had published, among other things, treatises on the social responsibility of business and the chart that now bears his name. More recently, techniques like the PERT (Project Evaluation and Review Technique) chart [35] have been trying to claim, thus far unsuccessfully, the Gantt chart's project management authority.

[5]Not too whimsical, though. Over $25 billion were spent by consumers in the USA on video games alone in 2009, and the company that produces the game in this example is worth about US$6 billion at the time of writing, so managing this industry is not just "for fun."

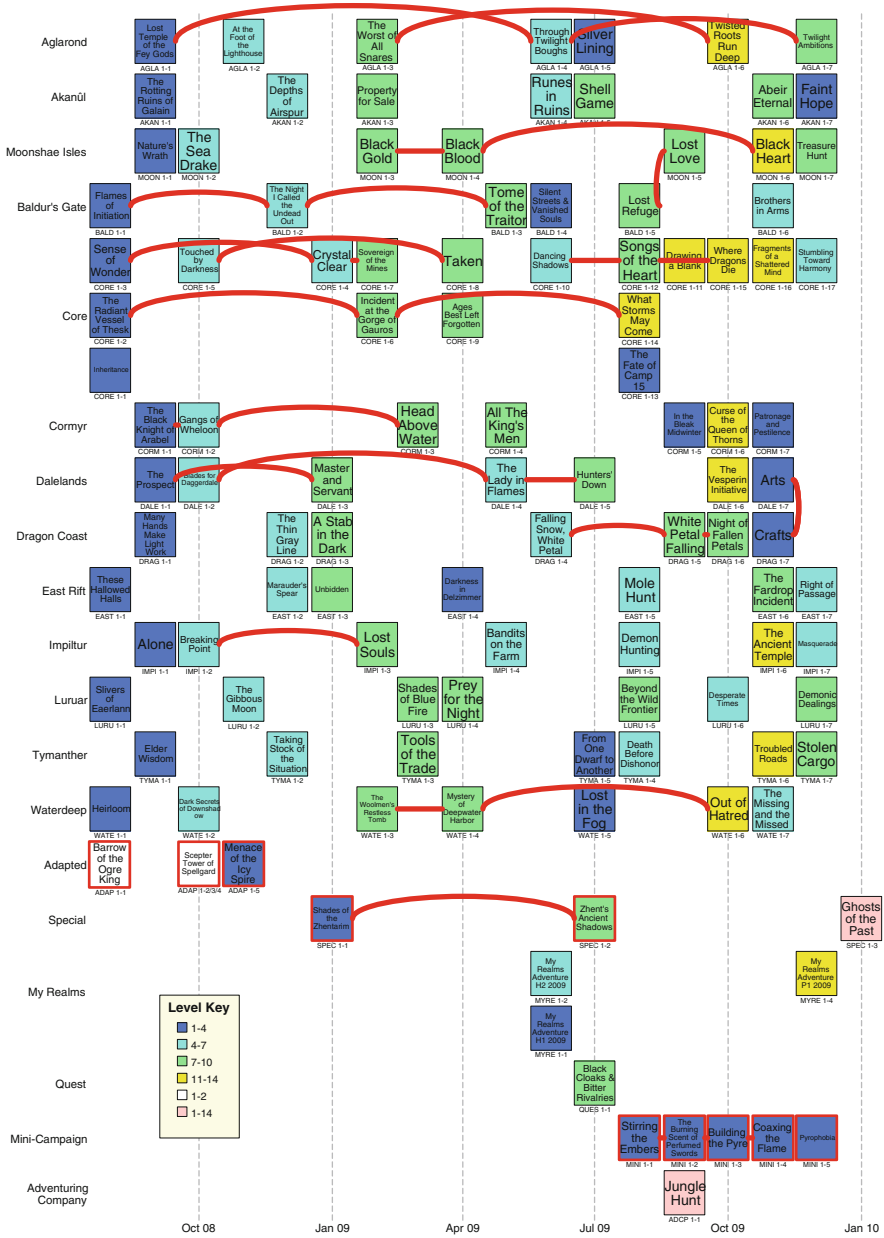


Fig. 10.8 Adventure series in a tabletop role-playing game. This figure encodes many pieces of information mainly in the *square point* elements. The *position* shows when the adventure module was made available for general play (x dimension) and what type of adventure it is (y dimension). The name of the adventure is shown within the element, with the code shown just below it as label aesthetics. The color of the adventure indicates which character levels it is suitable for, and a second color aesthetic defined on the *border of the box* highlights those adventures that are longer than the standard 4 hours. In the interactive version, pop-up *tool tips* give a long textual description for each adventure. Adventures that form a series are shown with *link elements* joining them into a chain

after a year of playing their first personas. Indeed, they added a new writing group mini-campaign that rapidly wrote five adventures that were longer than usual for low-level personas at that point, coinciding with the release of modules for personas of level 11 and higher. In this way they planned to appeal to established players in two ways (continuing their existing personas at high level and starting new ones at low level) as well as appealing to new players.

10.2.2 Timelines

Often in visualization, it is suggested that displays should be immediately obvious – there should be an "aha moment"; the information should "leap out at you." It is certainly true that if there is a strong feature of the data, and is not immediately obvious from the chart, then the visualization has a problem. For the types of data we are discussing in this section, though, the interest is in understanding the relationships between events and digging into the details. Figure 10.8 is complex because the goal is not a sudden understanding, but providing a visualization that can be searched for a strategy.

Timelines use time as one dimension, and then add information in a rich form at specific points or intervals along that line. As originally popularized in 1769 by Priestly [88], timeline charts simply showed the lifespans of famous people, with Fig. 10.9 as an example. This display is the fundamental timeline display and shows both the main strength and main weakness of the timeline.

The strength of the display is that it maps the important information – the spans of time – directly onto the horizontal dimension. This ensures that the reader can rapidly scan horizontally, to see how far apart in time two lifetimes are, and can also scan vertically, to see whose lifetimes overlapped (Demosthenes and Alexander, for example).

You have probably already noticed the weakness of the display; timelines need annotation, and those annotations take up a lot of the ink used for the display, and can be hard to read.[6] This figure is hand-drawn, and (looking ahead to an example we shall present later) the lettering in Fig. 10.11 is laid out by hand, allowing the

[6]Note also the difference in clarity between fonts used in the timeline figures in this section. The older, more ornate style is significantly harder to read. Naïvely we might think that the difference is between serif and sanserif fonts, but that is not necessarily true. Practically speaking different fonts are useful for different situations – the ornate font used in the title of Fig. 10.9 is fine, but the ornateness of the labeling hinders understanding. The documentary *Helvetica* [59] is a fascinating look at typefaces, and in particular the domination of the Helvetica font. Viewing the depth of feeling evidenced by typeface designers over the issue I am hesitant to recommend any particular approach, but Helvetica and its clone, Arial, do work out quite well in many situations, with Verdana another good choice. Arial and Verdana have the advantage of being designed for computer display and are available on both Macintosh and PC systems (but not Linux). Arial is more compact than Verdana, and so if you have to go for a single font for labels on a computer system, it is probably the best current default.

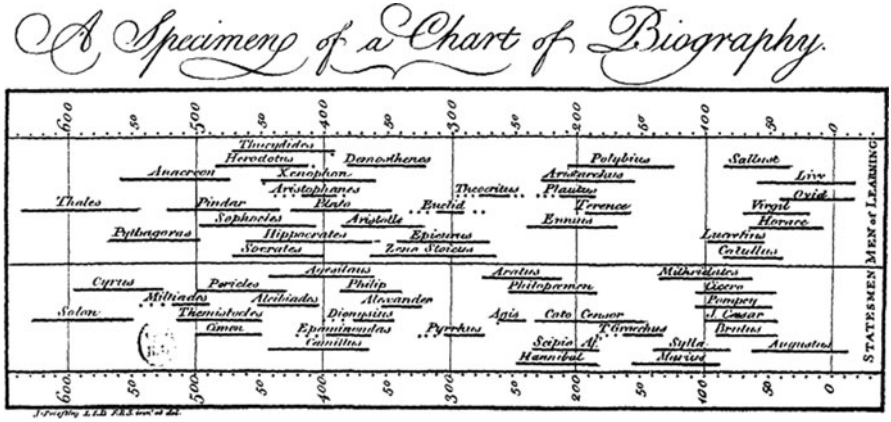


Fig. 10.9 An example timeline by Joseph Priestly, showing the time period leading up to 1 AD. Note the faceting into two sections, one on top for Men of Learning, and one below for Statesmen

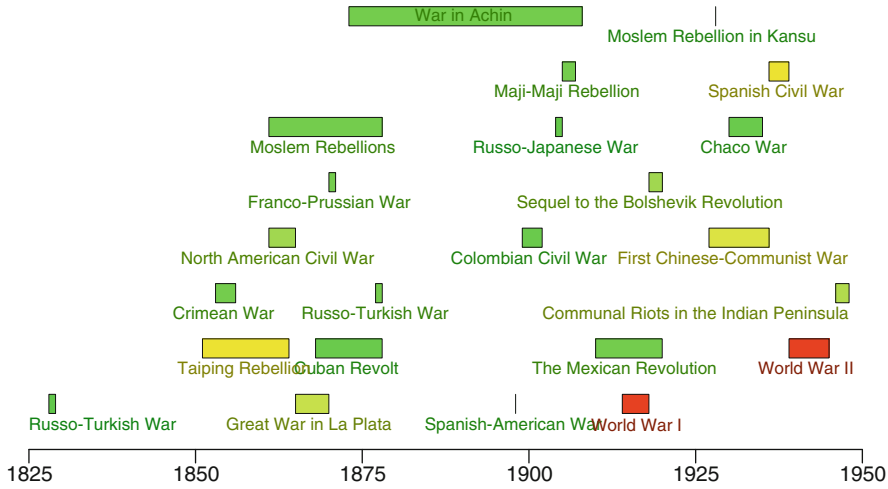


Fig. 10.10 The timeline here shows the spans of major wars over the timeline covered by Richardson's analysis of deadly quarrels. The color of the *interval element* and, redundantly, the *label* color indicate the magnitude of the war as measured by the number of casualties

designer to compensate for this problem. In this book we are trying to establish automatic rules and so Fig. 10.10 shows a more typical result, where an automatic label location algorithm has been used.

Figure 10.10 shows essentially the same timeline display, with the addition of a color aesthetic. The data shown are the statistics of deadly quarrels used in Chap. 5.

The use of color in this figure is worth a side note. Because the elements may be of very short span (e.g., the Spanish-American War), the color of the element might

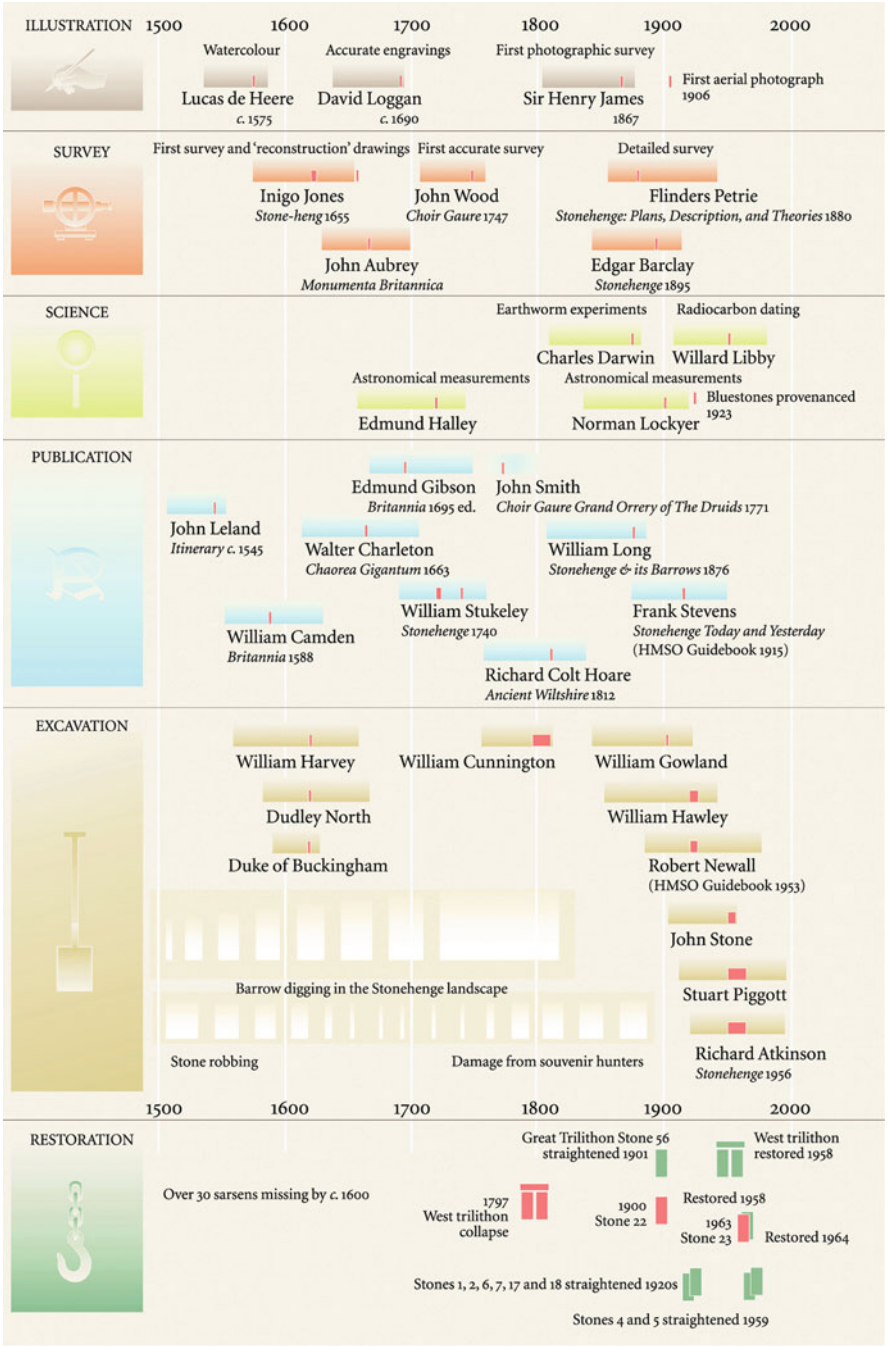


Fig. 10.11 Timeline of Stonehenge exploration. This figure is taken from the book *Solving Stonehenge: The New Key to an Ancient Enigma* [64], used with permission. A detailed explanation of the figure is given in the text

not be visible. Thus the label was used redundantly to encode the color. An astute observer will then note that although the hues of the elements and their labels match, the labels are darker than the elements themselves. This was done so as to ensure the text is legible on a white background (this page). Always be careful when designing visualizations to plan for extreme data, and ensure the chart will be useful in such circumstances. Again, comparison with Fig. 10.9 is informative. The variation in lengths of wars is much greater than the variation in lengths of lives of important people. It is rare to see a great man like Alexander, who lived less than 33 years; life spans for this chart have a min:max ratio of about 1:3. In contrast, some deadly quarrels are very quick (the Spanish-American war took 4 months), whereas some are longer (the Achinese War lasted from 1873 to 1904), for a ratio of about 1:100. Different statistical properties of the data lead to different design decisions.

In the two and a half centuries since their invention, countless versions of the timeline have been created, with many levels of detail and organization. The basic structure is of data ranges aligned on one dimension, labeled with information that allows an expert to apply their domain knowledge, and using aesthetics and faceting to add additional information. Figure 10.11 shows an excellent example of a timeline showing a wealth of concentrated information in a timeline format. The core data are coded as pastel-shaded rectangular ranges showing when various exploration activities took place, with the type of activity color coded and faceted to provide groupings. In addition to these base ranges for activities, multiple other elements are present:

Small red "subrange" elements. These show critical events that are often point events, and almost always lie within a range of the base element. One feature we can note is that the landmark events occur in the later half of the activity. As we might expect, papers and books are published when the work is nearing completion. Adding a second labeled element causes label layout issues, however, so the designer has mitigated this by using distinct typefaces for the two labelings, in much the same way a cartographer would use a different font for labeling regions as opposed to cities.

Low-information ranges. Although most activities are well documented, a considerable amount of unorchestrated digging and robbing took place over the earlier years, and these have been displayed with ranges that are shown in a style that makes them more of a background element than a major feature. A single label for a set of such low-information activities has been used, also deemphasizing their importance.

Icons. Restoration activities are not shown as ranges of times, but as simple events. Free from the constraints of showing a range, icons can be used to distinguish between simple stone straightening and trilithon fixing. Color is used to show bad events in red and good events in green.

This figure is a complex, hand-designed chart intended for reflective study. The design and layout invites us to make time comparisons, and help explain features of importance to researchers in this domain, such as answering questions like: What differences would we expect between Loggan's engravings and Sir Henry James'

photographs? As well as pattern discovery, this visualization has an important use as a reference: When we need to understand one person's contribution (such as Gibson's *Brittanica* article), we can see his contemporaries and parallel activities. This chart is valuable for both discovery and search goals.

10.3 Summary

Most of this book is targeted at presenting general data. In this chapter we have examined a couple of specific areas that need visualizations designed to highlight important features of this data. The examples were chosen to show the extremes in terms of data quantity. The section on large data shows how to apply the principles described in this book to data of arbitrarily large volume; the focus being on aggregation, filtering and showing features within the aggregation. The section on timelines shows a different extreme – very few items of data, but rich information on each item.

For large data, general aggregation techniques were shown, as well as techniques for augmenting standard displays to cope with large data volumes. One important lesson is that graphs and tables are not necessarily competing techniques for displaying summary data, but comprise different points along a continuum of possible visualizations. The *Grammar of Graphics* [135] provides building blocks for defining this continuum, as discussed in the framework Chap. 2.

For linked events and timelines very different techniques are required. The details are too important to be aggregated over, and the design goals become focused on maximizing the available information on single charts for reflective study. Issues of labeling and graphic design become less of an individual style preference and more important for enhancing clarity. These charts are not easy to design and may take many design iterations to get useful enough to use.

10.4 Further Exploration

The American Statistical Association site for this *Airline Data Expo* [1] contains a number of posters showing visualizations crafted to explore this data set. As with any such competition, it is valuable to look through them all and see what works and try and draw out the successful principles. It is also interesting to see how details of presentation make a big difference to clarity of results. Which axes drawing algorithms work best? Which fonts are better for labeling? Is it better to have a dark or a light background for a chart? By exploring a set of graphs for the same data set, we can learn more about what makes visualizations successful.

It is interesting to compare the Stonehenge exploration (Fig. 10.11) to those created by the *LifeLines* project [58, 86]. The latter figures are for a very different

domain, showing events in the treatment of health issues for medical patients, but the goals are similar and the end results show a remarkable similarity of design.

Linked event diagrams and timelines have a free dimension (in our examples and also typically, the vertical dimension). Items can be moved round within their faceting and should be placed so as to make best use of the space available, avoid overlap and, with linked events, avoid too many crossings. This is an important constrained layout problem, and the field of *graph layout* provides help on how to design algorithms to solve this problem. The Web site `http://graphdrawing.org` provides references to a number of books at various levels of complexity.

Chapter 11
Gallery of Figures

Time flies like an arrow. Fruit flies like a banana.

—Groucho Marx, (1890–1977)

11.1 Chart Complexity

For natural languages, a number of systems exist for classifying the effort required to understand pieces of text. They may be called "difficulty level," "reader level," "grade level," or some similar name, but the basic principle is the same: to be able to rank texts in order so that a reader who is comfortable with a certain level of complexity can tell which texts they will be able to read at the same level of comfort.

For visualization, we have a similar, but not identical, task. In visualization a good case can be made that if two charts convey the same information, then the simpler one (the one that can be understood more rapidly) is always to be preferred over the more complex one. In the language of visualization, function is much more important than form. A beautiful chart that is more complex than it needs to be is more suited for use as an art poster than a tool for making decisions. Would a web mapping application be better if it produced directions like the following?

> *While in the outermost lane of the historic Route 66, which terminates in the sun-baked hills of Santa Monica, smoothly steer in the direction of the setting sun, and as its glow warms your environs, keep a weather-eye out for a route variously known as "North Country Road" or "180." Shift over onto this road and remain steadfastly on it for a half score miles.*

Much importance is correctly attached to the usability of visualizations, which is strongly tied to their complexity. However, little work has been done on a general theory of chart complexity. In this book we have used a description of charts that

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2_11,

breaks them down into components: elements, coordinates, statistics, facets, etc. Intuitively it might be supposed that the more of these things are present in a graph, the more complex the result might be. In the spirit of such a supposition, a pilot experiment was performed. The goal of the experiment was to determine which grammatical factors led to a chart's being complex. The pilot study was designed to be *qualitative* – to inform as to which factors are worth considering and give a rough order of how important each grammatical component was. To this end, an experiment was designed that would ask a small set of people well trained in visualization to judge relative complexity as they perceived it. The results of this pilot study will be used in the future to design more thorough *quantitative* experiments to confirm and enhance the basic results discovered here.

11.1.1 Complexity Study Procedure

As a qualitative study, the basic procedure decided on was to provide people with a set of visualizations and have them rank the visualizations into their perceived order of complexity. The choice of judges was restricted to a set of people who were knowledgeable about visualization, people with knowledge of the grammar of graphics and ability to understand what a chart depicted without explanation.

The creation of the charts posed a significant problem;the goal was to choose a small number (20 was the number decided on) of charts from the billions of combinations of grammatical elements possible. Those charts should in some way encompass the range of possibilities and should also provide a variety of combinations of grammatical features, so that no two features should always be present together.[1] After some trial approaches, the following algorithm was adopted for choosing the charts:

1. The different chart combinations were chosen first by identifying, for each grammatical component, a small set of different possible aspects.
2. A random chart was defined as one where each grammatical element was selected randomly from the set of possible elements.
3. Ten million sets of 20 charts were generated, and the set that had the most two-way combinations of features was chosen as the most "diverse" for our purposes.
4. That final set was hand-edited to drop features from the charts if they would create invalid combinations (such as 2-D area charts with a size aesthetic).

[1]If every time we had a multi-element chart we also had faceting, it would then not be possible to tell which of the two was contributing to the perceived complexity. Hence we had the requirement that as many combinations should be present as possible.

Table 11.1 Experiment details: graphical components of each chart. This table lists the grammatical features of each chart used in the experiment. The subjects only saw the drawn charts together with their identifiers

ID	Elements	Aesthetics	Coordinates	Statistics	Facets	Scales
A	Point, surface	Shape	Rect-3	Summaries	2	Linear
B	Line	Color, size	Rect-2	None	None	Linear
C	Interval	None	Polar-2	Summaries	2-wrapped	Log-linear
D	Interval, line, point	Size, color	Rect-2	Summaries	2	Linear
E	Line	Dashing	Rect-2	Smooths	None	Log-log
F	Point	Size	Rect-1	None	2-wrapped	Linear
G	Area	None	Rect-2	Summaries	None	Linear
H	Interval	Color	Polar-2	Counts	2	Log-linear
I	Interval, line, point	None	Rect-2	Summaries	1	Linear
J	Interval, surface	Size	Rect-3	Summaries	1-wrapped	Log
K	Interval	Color, size	Rect-2	Summaries	None	Linear
L	Interval	Color, size	Rect-1	None	1	Linear
M	Area	Color	Rect-2	Counts	1	Linear
N	Interval	Color, brightness	Polar-1	Counts	None	Linear
O	Point	Size	Rect-2	Summaries	1	Log-linear
P	Area	None	Rect-3	Summaries	None	Linear
Q	Interval, point	None	Rect-2	Summaries	None	Linear
R	Line	Color	Rect-2	None	2-wrapped	Log-linear
S	Interval	None	Rect-1	None	None	Linear
T	Point	Size	Rect-2	None	None	Log-log

The charts were then built by hand to the desired specifications using a single data set as the source data. The grammatical features of each chart are listed in Table 11.1.

Figure 11.1 on the following page shows 4 of the 20 figures used in the experiment, reproduced here on a smaller scale. The figures used in the experiment were drawn to fit in a page of size 10 in. $\times$ 7.5 in., whereas the examples show have been drawn into a space of size 2.2 $\times$ 2.2 inches, and the resulting drawings have been reduced for reproduction here. Thus the originals presented to the subjects were laid out with more space and had readable fonts, legends, etc.

These figures were shown to a set of people who were knowledgeable about the graphical depiction of data and they were asked to rank the charts in order of complexity, from least complex to most complex. Each subject was given a short page of instructions, describing the goal and asking them to judge complexity based on the question: How hard would it be to make decisions based on a chart of this type? As part of the instructions, they were asked to ignore details of presentation and features specific to the data set being portrayed. The resulting orderings for the 20 charts are shown in Table 11.2.

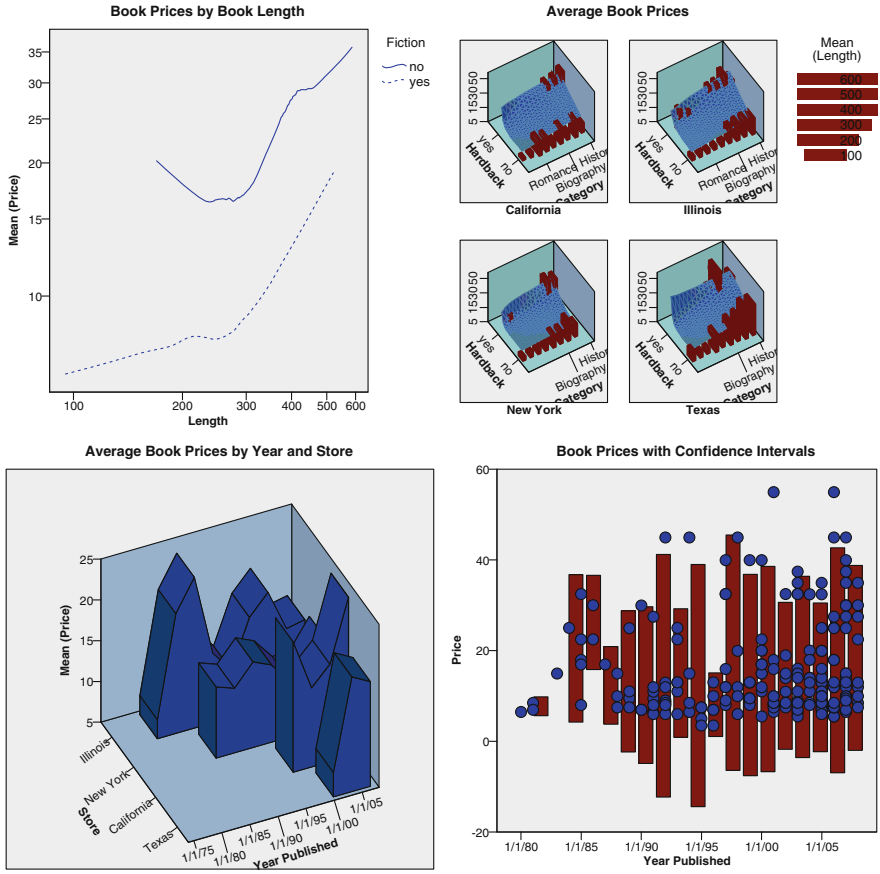


Fig. 11.1 Experiment visualizations. These visualizations show 4 of the 20 samples used in the pilot experiment:
(*top left*) Example **E**: *Lines* with a single aesthetic in rectangular 2-D with smooths on a log-log scale;
(*top right*) Example **J**: *Interval* with *size*, and surface in 3-D with log Y, in a single wrapped faceting;
(*bottom left*) Example **P**: *Area* using a *summary* statistic in 3-D;
(*bottom right*) Example **Q**: *Intervals* and *points* in 2-D, showing summaries.

11.1.2 Initial Analysis

The analysis of ranked data requires a fair amount of statistical expertise. Marden [75] provides an excellent resource for the general case and has been used as the main reference in this section.

An early issue in analysis of ranked data is that of the *distance* metric used – we need to have a measure of how different two rankings were. The distance measure

Table 11.2 Experiment results. Each row shows the subject's ordering of the perceived complexity of the charts

Subject	Ranking																			
1	G	S	E	N	T	Q	P	K	B	I	O	L	M	F	D	H	R	J	A	C
2	E	G	N	M	O	T	Q	S	I	D	A	P	F	R	K	L	J	C	H	B
3	G	E	N	M	S	I	O	T	Q	D	F	R	B	K	H	C	L	P	J	A
4	N	G	P	M	L	S	E	O	R	T	K	Q	I	J	D	F	C	H	B	A
5	E	G	T	R	M	N	P	O	D	Q	L	S	J	I	A	F	B	C	H	K
6	E	G	T	S	Q	M	O	R	F	D	I	N	H	C	K	L	P	J	A	B
7	E	T	N	G	M	S	Q	O	P	R	I	D	B	F	L	K	H	A	C	J
8	E	G	S	N	M	T	P	O	Q	K	L	D	R	F	H	C	I	B	A	J
9	G	E	S	N	M	O	Q	L	K	T	R	D	F	I	H	C	B	A	J	P
10	G	T	E	N	S	M	O	L	K	F	B	D	Q	I	R	C	H	P	A	J
11	G	E	S	N	B	T	C	M	Q	L	P	O	F	R	H	K	D	I	A	J

we use is Kendall's τ, which, when x_i is the rank given to item i by judge x and y_i is the rank given to item i by judge y, is defined as

$$\tau(x,y) = \sum_{i<j} I\{(x_i - x_j) * (y_i - y_j) < 0\}.$$

This distance, originally proposed in [66], measures the number of times the two rankings disagree over the relative order. It is often normalized to a unit scale by dividing by the possible disagreements (in our case that would be $20 \times 19/2 = 190$), but we have no need to do that in our analysis. Kendall's τ is a good choice for our study as it fits the desired goal: comparisons of relative complexity. The task of ranking the charts by perceived complexity can be thought of as a set of paired comparisons. Indeed, observation of the judges showed that this is what they did, taking a chart and comparing it to ones already ranked until the new one was placed in the correct location.

Armed with a suitable metric, we can now perform two useful tasks. We can compare one judge with every other judge and determine how different each judge is from the rest. We can also find the ordering that minimizes the overall distance to the judges' orderings. This is a measure of central tendency – a median or *consensus ordering* – giving the group consensus on the best ordering of the visualizations.

To calculate the best ordering there is no simple closed-form solution, and the naïve brute-force approach – trying all 20, or 2.43×10^{18}, different orderings – is not possible. Marden [75] gives a number of heuristics, and there are newer heuristics in the literature on this subject, but in our case a modified brute-force solution *was* possible. The algorithm used was as follows:

- Use the judge's ordering that has the least mean difference from the others as a starting point.

Table 11.3 Kendall τ distances. Each distance is an average of the distance from a given ranking to all the judges' rankings. The last line gives the distance for the consensus ranking

Subject	Mean distance
1	42.55
2	38.91
3	36.64
4	44.55
5	46.45
6	41.55
7	33.64
8	32.18
9	35.64
10	38.00
11	44.82
μ	29.37

- Try 1 million random swaps between pairs in the ranking, keeping any swapping that decreases the distance (and thus is a better candidate for the consensus ordering).
- Using that distance as the target to beat, run the search over the tree of 20 factorial possible orderings, but calculate the distances in a progressive manner. At a given depth of the search, when we choose a chart for the *n*th position, we know which other charts must be ranked after it, and so we can calculate the increase in the distance attributable to the choice at that given depth. This will give a minimum distance that any ordering starting with these n values must equal or exceed.
- If this partial distance calculated at a given depth exceeds our known best distance, abandon any further searching with that choice – there is no need to choose the rest of the sequence if the first n terms are already too different to be able to find a suitable choice.

The hope was that the initial guess would be good enough so that most choices in the search tree would be eliminated early on, leading to a vastly reduced search space. This hope proved true, and the consensus ranking given below was found with only 2.41×10^9 of the 2.43×10^{18} comparisons needed.[2] The consensus ranking found was

G E N S T M O Q P L R K D I F B H C A J

Table 11.3 gives the distances for both the judges and the consensus ordering, which we designate as μ.

[2]The 2.4 billion searches took about 5 minutes on a 1.8-GHz laptop, with the algorithm implemented in Java.

The table indicates that no judge was very different from the group, and it gives a bound on the best result we can expect from our analysis (a distance of 29.37) as well as a range of values that are as good as our visualization experts: between 32 and 46.

11.1.3 Model Fitting

Marden [75] gives a number of possible models that can be used in this situation to explore the data. However, our goal is not the rankings by themselves (in which case we might use the Babington–Smith model or submodels like Mallows, which are easier to work with). Instead we want to treat the rankings as a response and model how the various grammatical factors are related to the rankings.

Our goal is to build a model that, for a given set of grammatical values, gives a numeric value for complexity. We evaluate the model by comparing the ordering given by ranking our sample charts according to the predicted complexity with the judges' orderings. We are making the assumption that the ordering of the charts reflects a common underlying measure of complexity, and we wish to model that measure.

As a first step, we transformed the data into a simpler form for traditional analysis. We took each judge's evaluation of each chart as a data point (11 judges $\times$ 20 charts = 220 data points) and used a variety of traditional techniques to model the relationship between the judge's ranking of that chart[3] and the grammatical factors. Simple correlation analyses, neural nets, and linear regression resulted in similar results; the models were not very good (the regression had $R^2 = 0.43$, for example), and the generated rankings were worse than any of the judges. Most importantly, the fitted models looked wrong. The regression gave a negative contribution to the number of coordinate dimensions, for example, which would imply that a 3-D bar chart was simpler than a 2-D bar chart. Trying to predict the *rank* of each visualization was not a good surrogate for predicting their "head-to-head" comparisons.

On a positive note, the standard models did provide a list of the grammatical features that might be important in the analysis. These factors were as follows:

- *ElementCount* – The number of elements in the chart (point, line, ...)
- *AestheticCount* – The number of aesthetics in the chart (color, size)
- *CoordinateCount* – The number of dimensions in the base coordinate system (1, 2, or 3)
- *PolarFlag* – If the chart had a polar transformation
- *SummariesFlag* – If the chart used a summary such as a mean or median

[3]An *ordering* is a list such as "A C D B E." The *ranking* derived from that ordering assigns the position of the element within that list to that element, so that rank(A)=1, rank(B)=4, rank(C)=2, rank(D)=3, rank(E)=5.

- *CountsFlag* – If the chart used a count statistic
- *FacetCount* – How many faceting dimensions were used
- *WrappedFlag* – If faceted, this flag was true if the chart wrapped the faceting
- *ScaleFlag* – True if there was any modification to the scales – log or power transforms

Rather than continue with a standard method, the next technique used was to fit a regression model based on the discovered factors listed above, but rather than trying to predict the complexity rankings, the complexities for the model were used to generate an ordering of the charts and that ordering compared to the judges' orderings using the techniques of Sect. 11.1.2 above. A brute-force approach was used to find the best parameters for the linear model, as there is no known closed-form solution to the model.

This gave a model with a mean distance of 41.73, consistent with how well the judges were performing. The effect of dropping factors was investigated, and dropping the terms *PolarFlag*, *ScaleFlag*, *ElementCount*, *SummaryFlag*, and *CountsFlag* produced a much simpler model with an acceptable distance of 47.1. Dropping any other terms produced a much larger distance. Then, adding terms back in was attempted, and adding *ElementCount* back in improved the result to 43.55. Adding any more terms did not make much improvement and removing any made it much worse. As a final step, the model was allowed to use noninteger parameters and the parameter space around the integer parameter model was searched to find a better fit. After a few iterations, the resulting distance measure was 41.64. A nice feature of the resulting model is that the constant value is zero, so a chart with nothing in it has zero complexity.

This procedure is similar to stepwise regression: substituting the distance measure for a statistical measure of the proportion of variance explained. To make it more rigorous, we could have built a model (using one of the models in Marden, for example) and then used comparisons of nested models more rigorously to determine what differences in the distance measure should be considered "large." For our qualitative purposes, using the range of values provided by the judges gives us the ability to do the same process more simply and without the necessary assumptions that would be needed to create such a model.

11.1.4 Discussion

The resulting model for complexity is given below:

$$\text{Complexity} = \text{ElementCount} + 1.5 * \text{FacetCount} + 1.9 * \text{AestheticCount} + 2.1 * \text{WrappedFlag} + 3.4 * \text{CoordinateCount}$$

and it provides an ordering for the charts used in the experiment like this:

S G N Q E T L M O P I F B K H C R D A J

This model has been evaluated based on the same data used to fit it, so we do not have any fair measure of how good a model it really is. However, the terms found look reasonable, as the model conforms to known studies (such as ones that show that 3-D charts are hard to understand), and it also fits the data well. It might be surprising that polar transforms do not make charts harder to understand, and the difficulty of using wrapped facetings suggests that techniques like trellis displays are not a great idea, but the results are to be used as a rough guide only; the main use will be to develop a more rigorous study in the future.

11.1.5 Application to the Figures in This Book

In this book, the results are used as guidelines, and in this chapter they have been used to order the list of figures used in the book from least complex to most complex, as shown in the following section, where we provide a gallery of all figures shown.[4] Somewhat arbitrarily, photographs and other nonchart figures are placed at the end of the list. We could attempt to ascribe grammatical concepts to them, but the goal of this book is to understand how to effectively visualize data, not real-world items, and so the simplest method is to remove them from consideration.

Each figure has a categorical color aesthetic applied for the label, which indicates the main feature of interest in the chart. The color coding is as follows:

Coordinates	Figures with complex coordinate system transformations
Faceting	Figures that show one or more facets (panels)
Multi-element	Figures that combine multiple elements
Statistics	Figures that have nontrivial statistics
Regular	Figures that have no dominating feature
Noncharts	Photographs and other images that are not chartlike

To the left of the label area is the computed complexity of the figure, and to the right is the figure reference and page number. This gallery of figures is itself a visualization – it is left as an exercise to the reader to determine its complexity. The reader is also encouraged to look at the resulting ordering of figures and consider whether it seems correct. If the proof of the pudding is in the eating, not in the looking, then for visualization, the value of a chart is not how pretty it looks, but how easy it is to use. What do you think?

[4] To avoid getting overly self-referential, figures in this chapter have been excluded from the gallery.

11.2 The Gallery

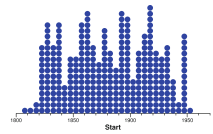
2.9 **0.1** (p.ix)

4.4 **1.1** (p.3)

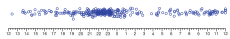
4.4 **1.2** (p.3)

4.4 **3.2** (p.70)

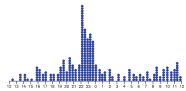
4.4 **3.7** (p.72)



4.4 **5.3** (p.109)

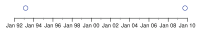


4.4 **5.6** (p.111)



4.4 **5.7** (p.112)

4.4 **6.8** (p.133)

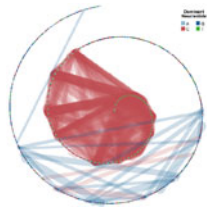


4.4 **6.9** (p.135)

4.4 **6.10** (p.136)

5.4 **5.12** (p.119)

6.3 **5.13** (p.120)

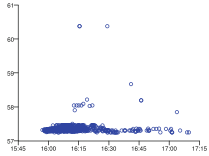


6.3 **6.6** (p.130)

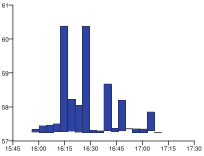
6.7 **7.5** (p.163)

Coordinates | Faceting | Multi-element | Statistics | Regular | Noncharts

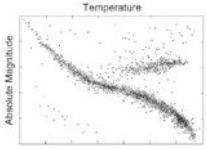
6.7 **7.6** (p.165)



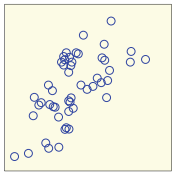
7.8 **2.1** (p.24)



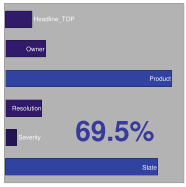
7.8 **2.8** (p.32)



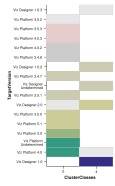
7.8 **3.1** (p.68)



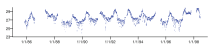
7.8 **3.13** (p.77)



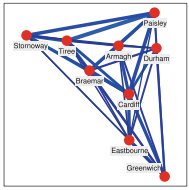
7.8 **3.16** (p.82)



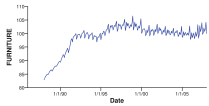
7.8 **3.19** (p.84)



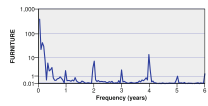
7.8 **4.2** (p.100)



7.8 **6.18** (p.148)

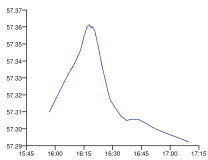


7.8 **8.4** (p.174)

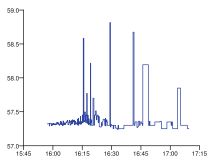


7.8 **8.5** (p.175)

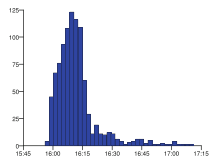
7.8 **10.9** (p.223)



7.8 **2.2** (p.25)

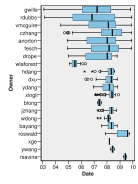


7.8 **2.3** (p.26)



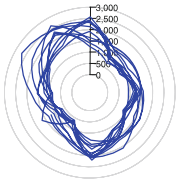
7.8 **2.6** (p.29)

Coordinates Faceting Multi-element Statistics Regular Noncharts

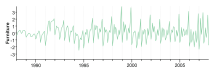


7.8 **3.12** (p.76)

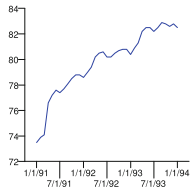
7.8 **4.3** (p.100)



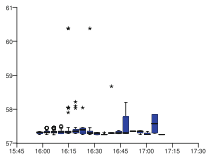
7.8 **6.4** (p.128)



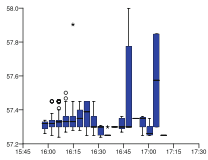
7.8 **9.7** (p.194)



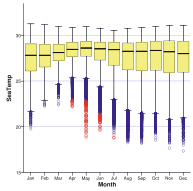
7.8 **9.14** (p.207)



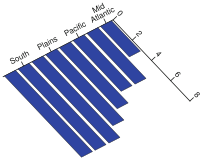
7.8 **2.9** (p.33)



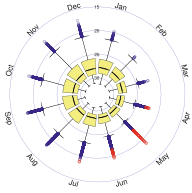
7.8 **2.10** (p.34)



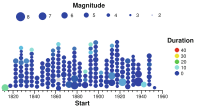
7.8 **2.29** (p.54)



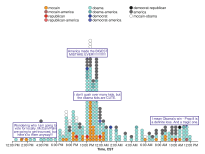
7.8 **2.27** (p.52)



7.8 **2.28** (p.54)

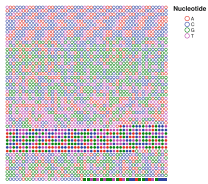


8.2 **5.4** (p.110)

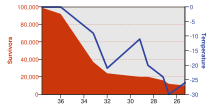


8.2 **5.8** (p.113)

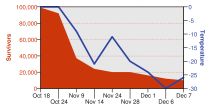
Coordinates | Faceting | Multi-element | Statistics | Regular | Noncharts



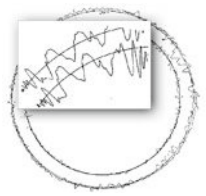
8.2 **5.11** (p.118)



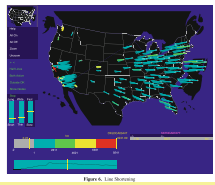
8.8 **1.8** (p.13)



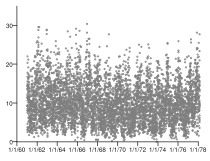
8.8 **1.9** (p.13)



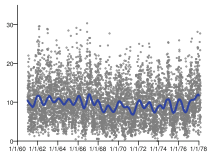
8.8 **5.2** (p.108)



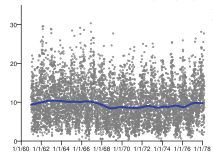
8.8 **10.7** (p.219)



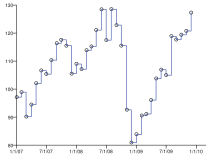
8.8 **2.12** (p.36)



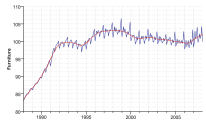
8.8 **2.13** (p.37)



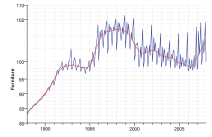
8.8 **2.14** (p.38)



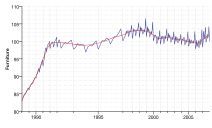
8.8 **8.6** (p.176)



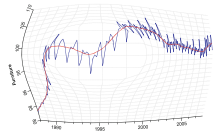
8.8 **9.3** (p.190)



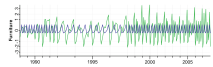
8.8 **9.4** (p.191)



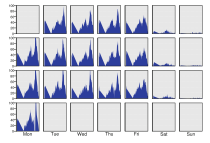
8.8 **9.5** (p.192)



8.8 **9.6** (p.192)

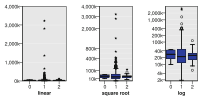


8.8 **9.8** (p.196)



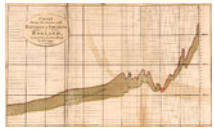
9.3 **6.13** (p.140)

Coordinates | Faceting | Multi-element | Statistics | Regular | Noncharts

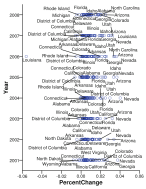


9.3 **9.9** (p.197)

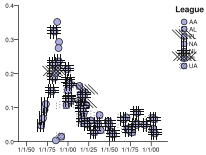
9.7 **1.4** (p.7)



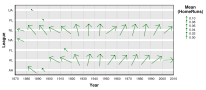
9.7 **1.5** (p.9)



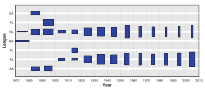
9.7 **1.12** (p.19)



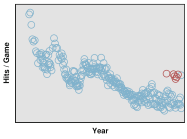
9.7 **2.19** (p.44)



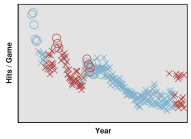
9.7 **2.20** (p.45)



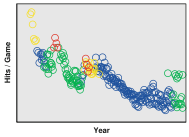
9.7 **2.21** (p.46)



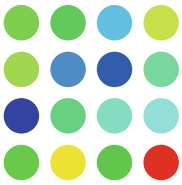
9.7 **2.22** (p.47)



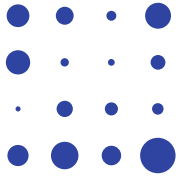
9.7 **2.23** (p.48)



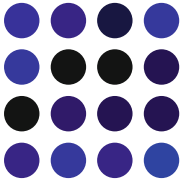
9.7 **2.25** (p.49)



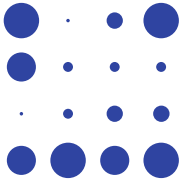
9.7 **3.3** (p.71)



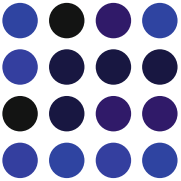
9.7 **3.4** (p.71)



9.7 **3.5** (p.71)

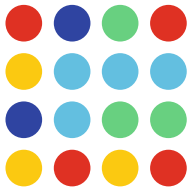


9.7 **3.8** (p.73)

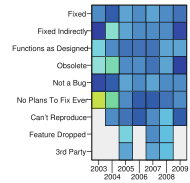


9.7 **3.10** (p.73)

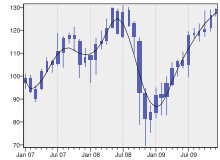
Coordinates | Faceting | Multi-element | Statistics | Regular | Noncharts



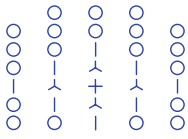
9.7 **3.9** (p.73)



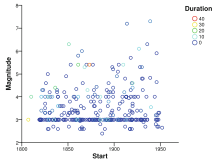
9.7 **3.11** (p.74)



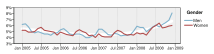
9.7 **3.14** (p.78)



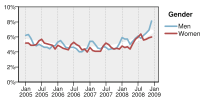
9.7 **4.1** (p.99)



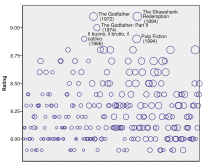
9.7 **5.5** (p.110)



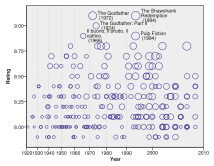
9.7 **6.2** (p.125)



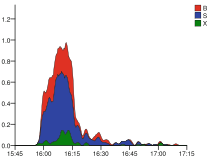
9.7 **6.3** (p.127)



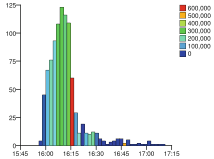
9.7 **8.2** (p.171)



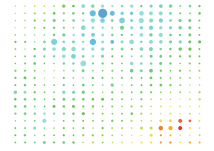
9.7 **8.3** (p.172)



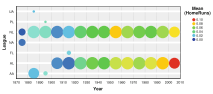
9.7 **2.4** (p.28)



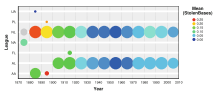
9.7 **2.7** (p.30)



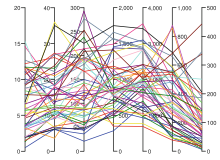
9.7 **2.16** (p.40)



9.7 **2.17** (p.42)

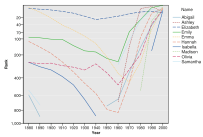


9.7 **2.18** (p.42)

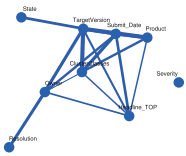


9.7 **2.26** (p.51)

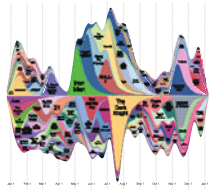
Coordinates | Faceting | Multi-element | Statistics | Regular | Noncharts



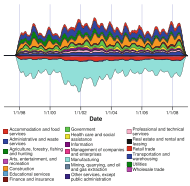
9.7 **3.15** (p.79)



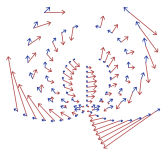
9.7 **3.17** (p.83)



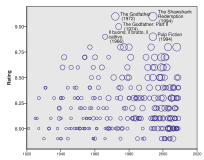
9.7 **5.9** (p.115)



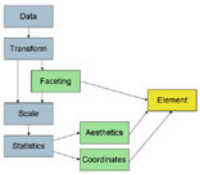
9.7 **6.1** (p.124)



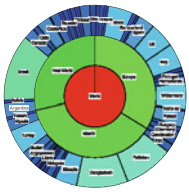
9.7 **6.7** (p.131)



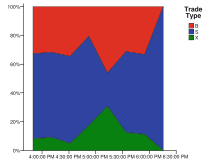
9.7 **8.1** (p.170)



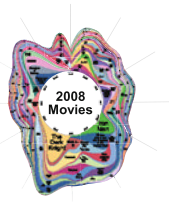
9.7 **9.1** (p.185)



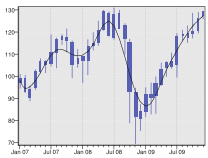
9.7 **9.11** (p.202)



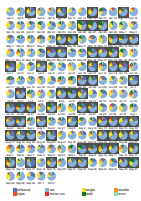
9.7 **2.5** (p.29)



9.7 **6.5** (p.129)

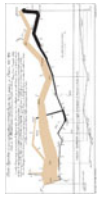


9.8 **8.7** (p.178)

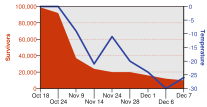


9.9 **6.14** (p.142)

Coordinates | Faceting | Multi-element | Statistics | Regular | Noncharts

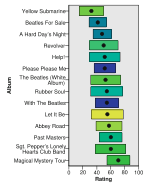


10.7 **1.6** (p.11)

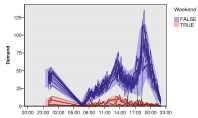


10.7 **1.7** (p.12)

10.7 **3.23** (p.91)



10.7 **3.21** (p.87)

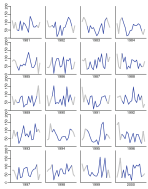


10.7 **6.12** (p.139)

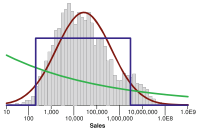
10.7 **10.6** (p.218)

10.8 **6.16** (p.145)

10.8 **6.17** (p.147)

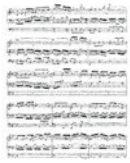


10.8 **6.19** (p.149)



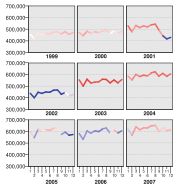
10.8 **3.6** (p.72)

11.2 **1.11** (p.18)

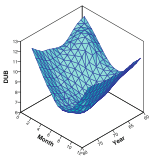


11.2 **5.10** (p.117)

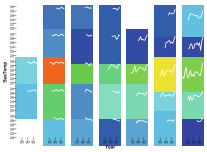
Coordinates Faceting Multi-element Statistics Regular Noncharts



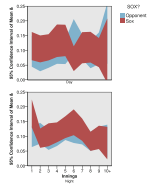
11.2 **10.5** (p.217)



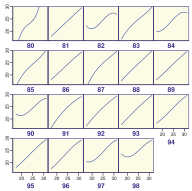
11.2 **2.15** (p.39)



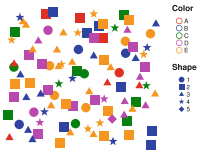
11.2 **2.30** (p.55)



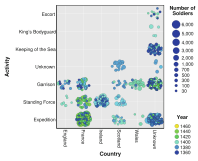
11.2 **6.11** (p.137)



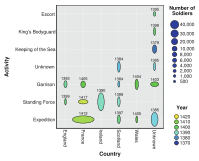
11.4 **2.31** (p.57)



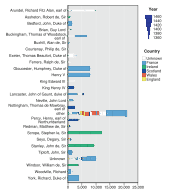
11.6 **2.24** (p.49)



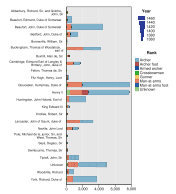
11.6 **7.1** (p.153)



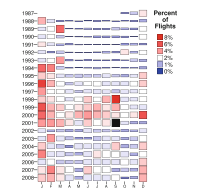
11.6 **7.2** (p.155)



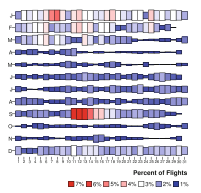
11.6 **7.3** (p.159)



11.6 **7.4** (p.160)

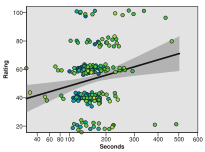


11.6 **10.2** (p.212)

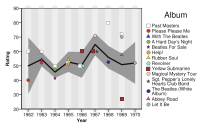


11.6 **10.3** (p.214)

11.6 **10.10** (p.224)

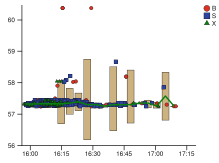


11.7 **3.20** (p.86)

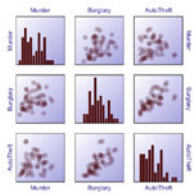


11.7 **3.22** (p.88)

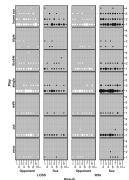
Coordinates | Faceting | Multi-element | Statistics | Regular | Noncharts



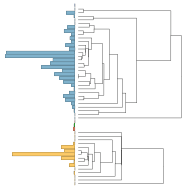
11.7 **2.11** (p.35)



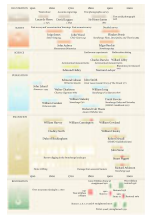
11.8 **2.33** (p.60)



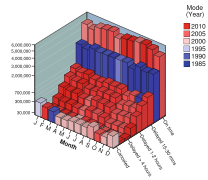
12.3 **6.15** (p.143)



12.6 **3.18** (p.83)

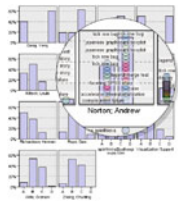


12.7 **10.11** (p.225)

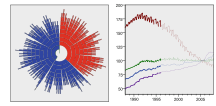


13.1 **10.1** (p.211)

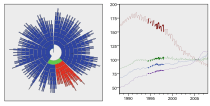
13.5 **10.4** (p.216)



14.1 **9.2** (p.188)



16.0 **9.10** (p.201)

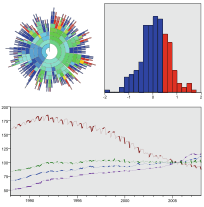


16.0 **9.12** (p.203)

16.4 **10.8** (p.221)

16.6 **1.10** (p.16)

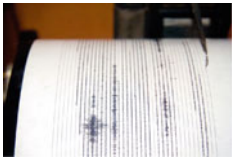
Coordinates | Faceting | Multi-element | Statistics | Regular | Noncharts



17.0 **9.13** (p.204)

2.32 (p.58)

1.3 (p.4)



5.1 (p.106)

References

1. ASA Sections on Statistical Computing and Statistical Graphics: Airline on-time performance. url: http://stat-computing.org/dataexpo/2009 (2009)
2. American Statistical Association: The American Statistical Association Section on Statistical Graphics. http://stat-graphics.org/graphics/
3. Basili, V.R., Caldiera, G., Rombach, D.H.: The Goal Question Metrics Approach. In: Encyclopedia of Software Engineering, vol. I, 1st edn., pp. 528–532. Wiley (1994)
4. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs, 1st edn. Prentice Hall, Upper Saddle River (1998)
5. Becker, R.A., Cleveland, W.S., Shyu, M.J.: The visual design and control of trellis display. J. Comput. Graph. Stat. **5**, 123–155 (1996)
6. Becker, R.A., Cleveland, W.S., Wilks, A.R.: Dynamic graphics for data analysis (c/r: p50-72). Dyn. Graph. Stat. **0**(0), 1–50 (1988)
7. Becker, R.A., Eick, S.G., Wilks, A.R.: Visualizing network data. IEEE Trans. Vis. Comput. Graph. **1**, 16–28 (1995)
8. Bertin, J.: Semilogie Graphique. Mouthon-Gauthiers-Villars, Paris (1967)
9. Bertin, J.: Semiology of Graphics. University of Wisconsin Press, Madison (1983)
10. Boehm, B., Rombach, H.D., Zelkowitz, M.V.: Foundations of Empirical Software Engineering: The Legacy of Victor R. Basili. Springer, Secaucus (2005)
11. Brewer, C.A.: Color use guidelines for mapping and visualization. In: A. MacEachren, D. Taylor (eds.) Visualization in Modern Cartography, chap. 7, pp. 123–147. Elsevier, Tarrytown, NY (1994)
12. Brewer, C.A.: Guidelines for use of the perceptual dimensions of color for mapping and visualization. In: Color Hard Copy and Graphic Arts III, Proceedings of the International Society for Optical Engineering (SPIE), San Jose, vol. 2171, pp. 54–63 (1994)
13. Brewer, C.A.: ColorBrewer. http://colorbrewer2.org/ (2003)
14. Brewer, C.A., Hachard, G.W., Harrower, M.A.: Colorbrewer in print: a catalog of color schemes for maps. Cartogr. Geogr. Inf. Sci. **30**(1), 5–32 (2003)
15. Buja, A., Cook, D., Swayne, D.: Interactive High-Dimensional Data Visualization. J. Comput. Graph. Stat. **5**(1), 78–99 (1996)
16. Card, S.K., Mackinlay, J., Shneiderman, B.: Readings in Information Visualization: Using Vision to Think. Series in Interactive Technologies. The Morgan Kaufmann, Waltham (1999)
17. Chatfield, C.: The Analysis of Time Series: An Introduction, 6th edn. Chapman & Hall/CRC, London, UK (2003)
18. Chen, C.h., Hrdle, W., Unwin, A.: Handbook of Data Visualization (Springer Handbooks of Computational Statistics), 1 edn. Springer TELOS, Santa Clara (2008)

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2,

19. Chen, J., MacEachren, A.M.: Supporting the process of exploring and interpreting space-time multivariate patterns: The visual inquiry toolkit. Cartogr. Geogr. Inf. Sci. **35**, 33–50 (2008)
20. Chernoff, H.: The use of faces to represent points in *k*-dimensional space graphically. J. Am. Stat. Assoc. **68**(0), 361–368 (1973)
21. Cleveland, W.: The Elements of Graphing Data. Hobart, Lafayette, IN (1985)
22. Cleveland, W.: Visualizing data. Hobart, Lafayette, IN (1993)
23. Cleveland, W.C., McGill, M.E.: Dynamic Graphics for Statistics. CRC, Boca Raton (1988)
24. Cleveland, W.S.: Graphical methods for data presentation: Full scale breaks, dot charts, and multibased logging. Am. Stat. **38**(4), 270–280 (1984)
25. Cleveland, W.S.: A model for studying display methods of statistical graphics (with discussion). J. Comput. Stat. Graph. **2**, 323–364 (1993)
26. Cleveland, W.S., Devlin, S.J.: Locally weighted regression: an approach to regression analysis by local fitting. J. Am. Stat. Assoc. **83**(0), 596–610 (1988)
27. Cleveland, W.S., McGill, M.E., McGill, R.: The shape parameter of a two-variable graph. J. Am. Stat. Assoc. **83**(402), 289–300 (1988)
28. Cleveland, W.S., McGill, R.: The many faces of a scatterplot. J. Am. Stat. Assoc. **79**(388), 807–822 (1984)
29. Cook, D.: The UCI KDD Archive. http://kdd.ics.uci.edu/; University of California, Department of Information and Computer Science
30. Cook, D., Swayne, D.F.: Interactive and Dynamic Graphics for Data Analysis: With Examples Using R and GGobi. Springer, Berlin Heidelberg New York (2007)
31. Cox, T.F., Cox, M.A.: Multidimensional Scaling. Chapman & Hall, London, UK (2001)
32. Craig, P., Haslett, J., Unwin, A., Wills, G.: Moving statistics - an extension of "brushing" for spatial data. In: Berk, Malone (eds.) Proceedings of the 21st Symposium on the Interface, pp. 170–174 (1989)
33. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Algorithms for drawing graphs: an annotated bibliography. Comput. Geom. Theory Appl. **4**(5), 235–282 (1994)
34. Donoho, A.W., Donoho, D.L., Gasko, M.: Macspin: Dynamic graphics on a desktop computer. IEEE Comput. Graph. Appl. **8**(4), 51–58 (1988)
35. Evarts, H.F.: Introduction to PERT. Allyn and Bacon series in quantitative methods for business and economics. Allyn and Bacon, Boston (1964)
36. Ewing, J.: On a new seismograph for horizontal motion. Trans. Seismol. Soc. Jpn. **2**, 45–49 (1880)
37. Few, S.: Show Me the Numbers: Designing Tables and Graphs to Enlighten. Analytics, Oakland (2004)
38. Few, S.: Now You See It: Simple Visualization Techniques for Quantitative Analysis. Analytics, Oakland (2009)
39. Fisherkeller, M., Friedman, J., Tukey, J.: Prim-s, an interactive multidimensional data display and analysis system. In: Dynamic Graphics for Statistics, pp. 91–109. Wadsworth, Pacific Grove (1975)
40. Fisherkeller, M.A., Friedman, J.H., Tukey, J.W.: Prim-9: An interactive multi-dimensional data display and analysis system. In: ACM Pacific, pp. 140–145 (1975)
41. Forta, B.: Sams Teach Yourself SQL in 10 Minutes, 3rd edn. Sams, Indianapolis (2004)
42. Friedman, J.: Exploratory projection pursuit. J. Am. Stat. Assoc. **82**, 249–266 (1987)
43. Friedman, J., Tukey, J.: A projection pursuit algorithm for exploratory data analysis. IEEE Trans. Comput. C **23**, 881–889 (1974)
44. Friendly, M.: Re-visions of minard. http://www.math.yorku.ca/SCS/Gallery/re-minard.html (2001)
45. Friendly, M.: Visions and Re-Visions of Charles Joseph Minard. J. Educ. Behav. Stat. **27**(1), 31–51 (2002)
46. Friendly, M., Denis, D.J.: Milestones in the history of thematic cartography, statistical graphics, and data visualization. http://datavis.ca/milestones/ (2008)
47. Gershon, N., Page, W.: What storytelling can do for information visualization. Commun. ACM **44**(8), 31–37 (2001)

48. Government, U.: UK Met Office. http://www.metoffice.gov.uk/
49. van Ham, F., Wattenberg, M., Viégas, F.B.: Mapping text with phrase nets. IEEE Trans. Vis. Comput. Graph. **15**(6), 1169–1176 (2009)
50. Haslett, J., Bradley, R., Craig, P., Unwin, A., Wills, G.: Dynamic graphics for exploring spatial data with application to locating global and local anomalies. Am. Stat. **45**(0), 234–242 (1991)
51. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer Series in Statistics. Springer, Berlin Heidelberg New York (2009)
52. Havre, S., Hetzler, E., Whitney, P., Nowell, L.: Themeriver: Visualizing thematic changes in large document collections. IEEE Trans. Vis. Comput. Graph. **8**(1), 9–20 (2002). DOI http://dx.doi.org/10.1109/2945.981848
53. Hayes, B.: Group Theory in the Bedroom, and Other Mathematical Diversions. MacMillan, New York (2009)
54. Hearst, M.A.: Search User Interfaces, 1 edn. Cambridge University Press, Cambridge, UK (2009)
55. Heer, J., Agrawala, M.: Multi-scale banking to 45 degrees. IEEE Trans. Vis. Comput. Graph. **12**(5), 701–708 (2006)
56. Holford-Strevens, L.: A Short History of Time. The Folio Society, London, UK (2007)
57. Huff, D.: How to Lie With Statistics. Norton, New York (1993)
58. Human-Computer Interaction Lab, University of Maryland: Lifelines for visualizing patient records. URL: http://www.cs.umd.edu/hcil/lifelines/ (1998)
59. Hustwit, G.: Helvetica. Documentary Video (2007)
60. Indulska, M., Orlowska, M.E.: On aggregation issues in spatial data management. In: ADC '02: Proceedings of the 13th Australasian database conference, pp. 75–84. Australian Computer Society, Darlinghurst, Australia (2002)
61. Inselberg, A.: The Plane with Parallel Coordinates. Vis. Comput. **1**, 69–91 (1985)
62. Inselberg, A.: Parallel Coordinates: Visual Multidimensional Geometry and Its Applications. Springer, Berlin Heidelberg New York (2009)
63. International Organization for Standardization: ISO 8601. Data elements and interchange formats — Information interchange — Representation of dates and times. International Organization for Standardization, Geneva, Switzerland (1988). See also 1-page correction, ISO 8601:1988/Cor 1:1991.
64. Johnson, A.: Solving Stonehenge: The New Key to an Ancient Enigma. Thames and Hudson, London, UK (2008)
65. Kaplan, A.: From Krakow to Krypton: Jews and Comic Books. Jewish Publication Society of America, Philadelphia (2008)
66. Kendall, M.: A new measure of rank correlation. Biometrika **30**(1/2), 81–93 (1938)
67. Kipling, R.: Jungle book. MacMillan, London, UK (1894)
68. Kohonen, T.: Self-Organizing Maps, *Springer Series in Information Sciences*, vol. 30. Springer, Berlin, Heidelberg (1995). (3rd extended edn. 2001)
69. Koike, K.: The Assassin's Road. No. 1 in Lone Wolf and Cub. Dark Horse, Milwaukie, OR (2000)
70. Kuhlthau, C.C.: Inside the search process: information seeking from the user's perspective. J. Am. Soc. Inf. Sci. **42**(5), 361–371 (1999)
71. Leung, Y.K., Aerley, M.D.: A review and taxonomy of distortion-oriented presentation techniques. ACM Trans. Comput.-Hum. Interact. **1**(2), 126–160 (1994)
72. Levkowitz, H.: Perceptual steps along color scales. Int. J. Imag. Syst. Technol. pp. 97–101 (1996)
73. Lie, H.W., Bos, B.: Cascading Style Sheets: Designing for the Web. Addison-Wesley Longman, Boston (1997)
74. Mallet, R.: Great Neapolitan Earthquake of 1857: The First Principles of Observational Seismology as Developed in the Report to the Royal Society of London of the Expedition Made by Command of the Society Into the Interior of the Kingdom of Naples, to Investigate the Circumstances of the Great Earthquake of 1857. Chapman & Hall, London, UK (1862)

75. Marden, J.I.: Analyzing and Modeling Rank Data. Chapman & Hall, London, UK (1995)
76. Martin, A., Ward, M.: High dimensional brushing for interactive exploration of multivariate data. In: Visualization, 1995. Visualization '95. Proceedings, IEEE Conference on, pp. 271–. Los Alamitos (1995)
77. McLachlan, R.: The earthquake. Nature **30** (1884)
78. Miller, J.E.: The Chicago Guide to Writing About Numbers. 0226526313. University of Chicago Press, Chicago (2004)
79. Minard, C.J.: Carte figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813. self-published (1861)
80. Mosteller, F., Tukey, J.: Data Analysis and Regression: A Second Course in Statistics. Addison-Wesley, Reading (1977)
81. Moyer, G.: Luigi lilio and the gregorian reform of the calendar. Sky and Telescope (1982)
82. Nightingale, F.: Diagram of the causes of mortality in the army in the East. Private publication (1858)
83. Ogata, Y.: Space-time point-process models for earthquake occurrences. Ann. Inst. Stat. Math. **50**(2), 379–402 (1998)
84. Orton, H., Sanderson, S., Widdowson, J. (eds.): The Linguistic Atlas of England. Routledge, New York (1978)
85. Pinker, S.: A theory of graph comprehension. In: R. Freedle (ed.) Artificial Intelligence and the Future of Testing (1990)
86. Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., Shneiderman, B.: Lifelines: Using visualization to enhance navigation and analysis of patient records. In: American Medical Informatic Association Annual Fall Symposium (Orlando, 9–11 Nov. 1998), pp. 76–80. AMIA (1998)
87. Playfair, W.: Commercial and Political Atlas: Representing, by Copper-Plate Charts, the Progress of the Commerce, Revenues, Expenditure, and Debts of England, during the Whole of the Eighteenth Century. Corry, London (1786)
88. Priestley, J.: A Description of a New Chart of History. J. Johnson, London, UK (1769)
89. Radford, M.: Photograph of stonehenge. http://www.flickr.com/photos/misterrad/ (2009)
90. Richard Weber, J.M.: Business intelligence competition, bi cup 2006. In: Talleres de Ingeniería de Sistemas 2007 (2006)
91. Richardson, L.: Statistics of Deadly Quarrels. Boxwood, Pittsburgh (1960)
92. Robbins, N.: Creating More Effective Graphs. Wiley-Interscience, New York (2004)
93. Russell, M.A.: Dojo: The Definitive Guide. O'Reilly Media, Sebastopol, CA (2008)
94. Sattler, K.U., Schallehn, E.: A data preparation framework based on a multidatabase language. In: IDEAS '01: Proceedings of the International Database Engineering & Applications Symposium, pp. 219–228. IEEE Computer Society, Washington, DC (2001)
95. Schervish, M.J.: P values: What they are and what they are not. Am. Stat. **50**(3), 203–206 (1996)
96. Schuster, A.: On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena. Terr. Magn. Atmos. Electr. **3**, 13–41 (1898)
97. Scott, D.W.: On optimal and data-based histograms. Biometrika **66**(0), 605–610 (1979)
98. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: IEEE Visual Languages, UMCP-CSD CS-TR-3665, pp. 336–343. College Park (1996)
99. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman & Hall, London, UK (1986)
100. Slingsby, A., Dykes1, J., Wood, J.: Using treemaps for variable selection in spatio-temporal visualisation. Inf. Vis. **7**, 210–224 (2008)
101. Smith, E.E.: Triplanetary. Amazing Stories **Jan - Apr** (1934)
102. Sobel, D.: Longitude. Penguin, London, UK (1995)
103. Sobel, D.: Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time. Penguin (1996)
104. Software, T.: Tableau software. url: http://www.tableausoftware.com

105. Stevens, S.S.: On the theory of scales of measurement. Science **103**, 677–680 (1946)
106. Stoker, B.: Dracula. Constable, London, UK (1897)
107. Sullivan, L.H.: The tall office building artistically considered. Lippincott's Mag. (1896)
108. Swayne, D.F., Cook, D., Buja, A.: XGobi: Interactive Dynamic Graphics in the X Window System with a Link to S. In: American Statistical Association 1991 Proceedings of the Section on Statistical Graphics, pp. 1–8. American Statistical Association, Alexandria (1992)
109. Theus, M.: Trellis displays vs. interactive graphics. Comput. Stat. **10**, 113–127 (1995)
110. Titchener, E.B.: Experimental Psychology: A Manual of Laboratory Practice. University of Michigan Press, Ann Arbor (1901)
111. Tufte, E.R.: The visual display of quantitative information, 2 edn. Graphics Press, Chesire, CT (2001)
112. Tukey, J.W.: Exploratory data analysis. Addison Wesley, Boston (1977)
113. Tukey, J.W., Tukey, P.: Strips displaying empirical distributions: 1. textured dot strips. Tech. rep., Bellcore (1990)
114. University of Reading: The Soldier in Later Medieval England Online Database. url: http://medievalsoldier.org (2009)
115. Unwin, A., Volinsky, C., Winkler, S.: Parallel coordinates for exploratory modelling analysis. Comput. Stat. Data Anal. **43**(4), 553–564 (2003)
116. Unwin, A.R., Hawkins, G., Hofmann, H., Siegl, B.: Interactive Graphics for Data Sets with Missing Values - MANET. J. Comput. Graph. Stat. **5**(2), 113–122 (1996)
117. U.S. Government: Popular Baby Names. http://www.ssa.gov/babynames
118. Velleman, P.F.: Data desk. The New Power of Statistical Vision. Data Description Inc (1992)
119. Velleman, P.F., Wilkinson, L.: Nominal, ordinal, interval, and ratio typologies are misleading (c/r: 93v47 p314-316; com: 94v48 p61-62). Am. Stat. **47**(0), 65–72 (1993)
120. Viégas, F., Wattenberg, M.M.: ManyEyes. http://manyeyes.alphaworks.ibm.com (2007)
121. Viegas, F.B., Wattenberg, M., van Ham, F., Kriss, J., McKeon, M.: ManyEyes: a Site for Visualization at Internet Scale. IEEE Trans. Vis. Comput. Graph. **13**(6), 1121–1128 (2007). DOI 10.1109/TVCG.2007.70577
122. Wainer, H.: Graphic Discovery: A Trout in the Milk and Other Visual Adventures. Princeton University Press, Princeton (2004)
123. Wainer, H., Spence, I. (eds.): The Commercial and Political Atlas and Statistical Breviary. Cambridge University Press, Cambridge, UK (2005)
124. Ware, C.: Visual Thinking for Design, illustrated edn. Morgan Kaufmann, Waltham (2008)
125. Wattenberg, L.: Baby Name Wizard. http://www.babynamewizard.com/voyager
126. Wattenberg, M.: Arc diagrams: visualizing structure in strings. In: Information Visualization, 2002, pp. 110–116 (2002)
127. Wattenberg, M.: Baby names, visualization, and social data analysis. In: IEEE Symposium on Information Visualization (InfoVis 2005). IEEE Computer Society, Los Alamitos (2005)
128. Wegman, E.: Hyperdimensional Data Analysis Using Parallel Coordinates. J Am. Stat. Assoc. **85**, 664–675 (1990)
129. Wegman, E.J.: Hyperdimensional data analysis using parallel coordinates. J. Am. Stat. Assoc. **85**(411), 664–675 (1990)
130. Wickham, H.: ggplot: an implementation of the grammar of graphics in r. In: UseR Conference, Vienna (2006)
131. Wickham, H.: ggplot: Elegant Graphics for Data Analysis. Springer, Berlin Heidelberg New York (2009). ISBN: 978-0-98140-6
132. Wilhelm, A.: Interactive Statistical Graphics: The Paradigm of Linked Views. Universität Augsburg, Augsburg, Germany (1999)
133. Wilkinson, D.O.: Deadly Quarrels: Lewis F. Richardson and the Statistical Study of War. University of California Press, Berkeley (1980)
134. Wilkinson, L.: The Grammar of Graphics. Statistics and Computing. Springer, Berlin Heidelberg New York (1999)
135. Wilkinson, L.: The Grammar of Graphics. Statistics and Computing. Springer, Berlin Heidelberg New York (2005)

136. Wilkinson, L., Anand, A., Grossman, R.: Graph-theoretic scagnostics. In: IEEE Symposium on Information Visualization, pp. 157–164. IEEE Computer Society, Los Alamitos (2005)
137. Wilkinson, L., Wills, G.: Scagnostics distributions. J. Comput. Graph. Stat. **17**(2), 473–491 (2008). DOI 10.1198/106186008
138. Wills, G.: Linked data views. In: A.U. Chun-houh Chen Wolfgang Härdle (ed.) Handbook of Data Visualization (Springer Handbooks of Computational Statistics), chap. II.9, pp. 216–241. Springer, Santa Clara (2008)
139. Wills, G., Wilkinson, L.: Autovis: automatic visualization. Inf. Vis. **9**, 47–69 (2010)
140. Wills, G.J.: Natural selection: Interactive subset creation. J. Comput. Graph. Stat. **9**(3) (2000)
141. Wills, G.J., Keim, D.: Data visualization for domain exploration: interactive statistical graphics. In: Handbook of Data Mining and Knowledge Discovery, pp. 226–232. Oxford University Press, Inc., New York (2002)
142. Wright, W.: Simcity. Computer Game (1988)
143. Young, F.: Visual Statistics: Seeing Data with Dynamic Interactive Graphics. Wiley-Interscience, New York (2006)
144. Young, F.W., Hamer, R.M.: Multidimensional Scaling: History, Theory, and Applications. Erlbaum, Mahwah, NJ (1987)

Index

G. Wills, *Visualizing Time: Designing Graphical Representations for Statistical Data*, Statistics and Computing, DOI 10.1007/978-0-387-77907-2,