Au: Please check chapter cross-refs (highlighted)

6

Visualizing Security Data

"The human visual system is a pattern seeker of enormous power and subtlety. The eye and the visual cortex of the brain form a massively parallel processor that provides the highest bandwidth channel into human cognitive centers."

Colin Ware, ***Information Visualization***

OK

Chapter 1 briefly mentioned that data analysis is similar to how archeology might be: spending hour after hour with small tools in the hope of uncovering even the tiniest of insights in the earth. That analogy can be extended into the shared desire to create a narrative. Archeologists attempt to recreate the stories of history by digging up parts of a story; it's the same with data analysts. There are stories buried in the data and it's up to the data analyst to uncover that narrative, piece it back together, and communicate that story to others. When it comes to data, with its unique blend of complexity and subtlety, nothing can tell a good story—a ***data story***—like a well-crafted visualization.

semicolon

A data story is built from several attributes, the two most important of which are ***truth*** and ***relevance***. Although you can have a good story without truth, you cannot have a good ***data story*** without truth. You cannot affect meaningful and successful change if your stories are built on lies or half-truths. Therefore, you need all the skills to uncover the truth within the data, and then you need the visualization skills to be sure the story the reader perceives matches the story you uncovered. The visual language should be a wrapper around the truth; thus it needs to be clear and unambiguous. Every point, line, color, and shape you place into a visualization should carry some piece of information supporting the truth in the data and in the data story.

comma

A good story is good only when it is relevant and actionable to the reader. You wouldn't want to show a board-level executive the Security Incident & Event Management (SIEM) dashboard any more than you'd want to force market reports on the SIEM operator. Stories fail to communicate if the readers don't feel they apply to them. Therefore you have to know the audience for your visualizations. Are you trying to elicit a budget change or firewall change? As you create your message, a good question to ask yourself is "so what?" and if you struggle to answer that question for the reader, rethink the approach. Another good mental exercise is to run through a few other possible outcomes of the story. If the result of the visualization is the same (from the reader's perspective), you should be rethinking the visualizations. For example, if you're showing a line graph with an obvious upward slant, imagine if that line went down. Would the reader have a different reaction? If it went up much more than it does, ***so what***?

We aren't suggesting that all data should be visualized. If the story in the data is best summarized with a sentence in an email, so be it. If the data can be expressed in a simple lookup table, so be it. The goal here is communicating the data. If you can communicate better, more succinctly, or simpler in any other way, you should go with that method. We also aren't suggesting that visualizations be the center of the story. All data exists within a context and all our stories need to have a beginning, middle, and end. Visualizations can play an important and supporting role in the entire communication process, but it should not be the only means of communication. Your focus is on the successful communication of the narrative and the method of communication is just a means to that end.

comma

Why Visualize?

By far, the most efficient path to human understanding is through the visual sense. Like a good hacker, you need to learn about the system, understand how it functions (or why it doesn't function), and then exploit this cognitive system to achieve your goal. In this case, the goal is to effectively and efficiently communicate the stories you find in the data. There are many advantages to using data visualization as a communication tool compared to other methods. To paraphrase Colin Ware (who we quoted to open this chapter), data visualization has the following advantages:

- **Data visualizations communicate complexity quickly.** Descriptive statistics (mean, median, variance, and so on) exist to describe and simplify data but tend to remove subtleties that exist. It's possible to communicate millions of data points in seconds while minimizing the loss of detail and resolution through visualization.
- **Data visualizations enable recognition of latent patterns.** Patterns that would never be apparent using statistical methods or scanning the data may be revealed through visualization. When data is visually presented, patterns in a single variable or relationships across many variables may leap off the screen.
- **Data visualizations enable quality control on the data.** Mistakes and errors in data collection or preparation can often be revealed through visualization. Data visualizations can serve as a good and quick sanity check on your work.
- **Data visualizations can serve as a muse.** It's been said that most breakthroughs in science didn't start with a "Eureka!" but instead with a "Huh, that's odd." Laying out the data visually can give you a new perspective and help facilitate your thinking and discovery processes.

Unraveling Visual Perception

The human system for processing visual information is incredibly complex and much of our knowledge around it is still evolving. There are a few key (and hopefully easy) concepts that you should understand since knowing how the brain visually processes information will help you create great visuals. Equally as important, knowing this information will also help you avoid creating visuals that aren't effective or helpful.

Our eyes convert visual stimulus in the form of light into electrical signals for our brains. This information passes through stages of our ***visual memory***, each with a specific set of strengths, limitations, and functions. Before we are consciously aware of it, our brains rapidly scan the visual field, which is called ***preattentive processing***. Finally, the brain will instruct the eyes to focus elsewhere, and through a series of ***saccadic movements***, our eyes will focus on various features to help build the image in our mind. The goal is to use three concepts from our visual processing system to create a solid foundation for good visuals and dashboards.

stimuli / ?
either sounds okay I think

Visual Thinking

This section steps through the various stages of memory within our visual perception: iconic memory, working memory, and long-term memory.

- **Iconic memory** is the first stop for the visual information. It is a very brief stop, lasting around half a second or until new information comes in. What happens in this tiny window is critical to creating good visualizations and dashboards. Using the information stored in iconic memory, the brain preprocesses the image prior to giving it any conscious attention. From an evolutionary perspective, this is quite helpful; this preattentive processing can help you quickly identify possible threats in your environment. For example, anyone who has been driving when an animal dashes in front of the car has probably felt that urgent message from the brain when it recognizes a possible threat. We begin to react immediately even before we can process the full extent of the threat. Even though you don't want your visualizations to be treated like a threat, through the use of colors, shapes and other cues, you can leverage this visual searching and preattentive processing in order to draw attention and communicate some basic attributes of your data. This will make processing much easier when viewers begin to consciously process it, and we will discuss preattentive processing in detail later in this chapter.

- **Working memory** is the next stop and things get a little more complicated here. First the brain groups visual aspects into meaningful objects and holds these in working memory. There is a lot of flexibility within working memory. We can rapidly replace or drop objects as we take in more information, but this flexibility comes at a cost in capacity. We can hold only three to five objects in working memory depending on the task and objects. This limit is important when you are designing visualizations and dashboards. If you create a visualization with a legend that has 10 different attributes, viewers will have to continually reference the legend in order to understand what they're looking at. Therefore, as you communicate the stories in your data, limit each visual to no more than five objects (or four to be safe).
- **Long-term memory** is not directly involved in the visual processing but instead affects visual communication through the expectations and norms built up in long-term memory. In order for something to move into long-term memory, the viewer needs to visually "rehearse" the information to transition it from working memory into long-term memory. If the reader has seen visualizations before (and chances are very good they have), they have a certain level of expectation for what they are looking at. For example, if you create a scatterplot, the reader expects the origin of the graph to be in the lower left corner, with positive values of each axis extended up and to the right. If multiple colors are used, the reader will expect meaning to the color and will seek it out. It's very important to know what those norms and expectations might be, and if you deviate from them, do so for a very good reason and give visual queues to help people understand those deviations.

Tracking Eye Movements

When people focus on something like a dashboard or graphics on a computer screen, they do not simply fix their gaze on it and take in the image as a whole. Their eyes actually dash around the screen, focusing on very small portions for very short periods of time in order to build the image in their mind. These rapid eye movements are called ***saccades***, and overall they are called saccadic movements. They are anything but random. The brain has a set of rules (guidelines really) for how the next fixation point is prioritized. As an example, when another person greets you, your eyes perform scanning saccades over their entire face, bouncing from the distinct features of the face (eyes, nose, and mouth) and establishing the edges. The scanning saccades help you recognize not only the person, but also cues to allow you to judge their emotions.

The same applies to visualizations and dashboards. The eyes will fixate on an obvious feature and bounce around and between the points it considers important. Viewers build up the entire picture over a series of these movements and over time. Understanding these movements can help you build a visualization flow that seems natural (or at least not strained).

The saccadic motion is largely unconscious and is thought to be a ***ballistic*** movement. Once the brain initiates a saccadic movement, the muscles take over and handle the rapid acceleration and deceleration from beginning to end. This is important for two reasons—once it is initiated it cannot be changed or stopped and during the motion we suppress much of the visual input. We will want to limit the distance of these motions by creating compact dashboards and visualizations.

We can pull together a few important learning points from saccadic eye movements. Knowing that the eyes will bounce around from feature to feature and understanding the ballistic nature of the movement, you should keep several points in mind as you create dashboards and graphics:

- **Don't overload the dashboard with visual features.** Keep the number of attention-grabbing features under control because if everything is important visually, nothing will be important visually, and the reader will have to put more effort into understanding the visual.
- **Make the important messages obvious visual features.** Just as we scan the important parts of a human face, we look for the similar attention-grabbing features on the screen. Make sure that those features are clear and important to the viewer.
- **Limit time wasted on saccadic movements.** Saccadic movements that jump longer distances take longer to execute. Do not push the visual features into the corners or toward the edges. Forcing the viewer to bounce across large distances will decrease the amount of time they are actually seeing the features (and increase the time spent in saccadic movements).

The role of saccadic movements is more significant in the design of dashboards than with static data visualizations. A static visualization will typically have one, perhaps two, visual features we want draw attention to and the eye movements are contained in a relatively compact space. A dashboard may be designed to communicate several independent messages simultaneously with varying degrees of urgency. Good dashboard design, as you'll see in Chapter 10, limits the time viewers spend in a saccadic movement and exploits eye movements for efficiency.

that / to

comma

Preattentive Processing

The best way to describe preattentive processing is through pictures. Take a look at Figure 6-1 and try to count how many capital Xs there are in this completely random mix of letters and numbers.

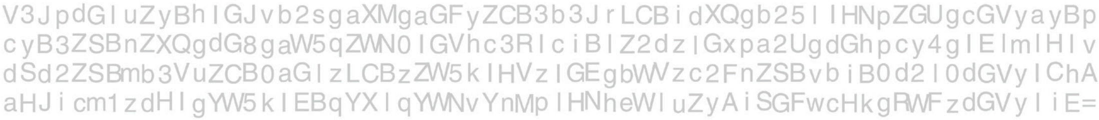

V3JpdGluZyBhIGJvb2sgaXMgaGFyZCB3b3JrLCBidXQgb25lIHNpZGUgcGVyayBp
cyB3ZSBnZXQgdG8gaW5qZWN0IGVhc3RlciBIZ2dzIGxpa2UgdGhpcy4gIElmIHIv
dSd2ZSBmb3VuZCB0aGIzLCBzZW5kIHVzIGEgbWVzc2FnZSBvbiB0d2I0dGVyIChA
aHJicm1zdHIgYW5kIEBqYXIqYWNvYnMpIHNheWIuZyAiSGFwcHkgRWFzdGVyIiE=

Figure 6-1 *Count the number of "X" characters*

Author: can we darken this image or is it okay as is?

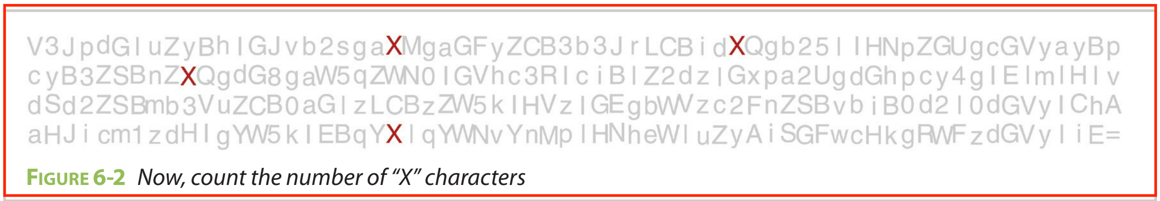
Do not darken, this color is part of the point.

Because all of the letters are the same color and contain the same relative space, nothing about any of the characters really stands out. The brain simply sees a collection of shapes. In order to count the Xs, you have to scan through each letter across the four rows. While you're doing that you also have to remember how many you've found so far. In contrast, look at a completely random mix of letters and numbers with the X characters emphasized (see Figure 6-2).

You can immediately see the Xs and count four of them. When you first look at this, your brain sees a background of gray symbols with four completely different objects that are similar to each other. Your preattentive processing mentally creates two groups: one of all the gray symbols and a second with the

Au: darken image?

Do Not Darken. The Dark "X" on here are the point



```
V3JpdGluZyBhIGJvb2sgaXMgaGFyZCB3b3JrLCBidXQgb25lIHNpZGUgcGVyayBp
cyB3ZSBnZXQgdG8gaW5qZWN0IGVhc3RlciBlZ2dzIGxpa2UgdGhpcy4gIElmIHlv
dSd2ZSBmb3VuZCB0aGlzLCBzZW5kIHVzIGEgbWVzc2FnZSBvbiB0d2l0dGVyIChA
aHJicm1zdHIgYW5kIEBqYXlqYWNvYnMpIHNheWluZyAiSGFwcHkgRWFzdGVyIiE=
```

Figure 6-2 *Now, count the number of "X" characters*

Comp: Move figure beneath para.

dark red Xs. A split-second later, you will consciously recognize the second group as what you're interested in (the Xs). It becomes trivial to visually exclude the gray characters and now you can scan just through this group. Counting the Xs becomes a simple and quick task.

That mental grouping and ease of focus is what you're after. You want to enable your preattentive processing to effortlessly group similar objects and highlight where you want attention to be focused. But you have to keep in mind that the preattentive processing is not all that smart. It cannot project meaning, interpret the objects, or make meaningful associations (beyond simple visual grouping).

Through hundreds of studies, researchers have been able to differentiate between visual attributes based on those that can be identified preattentively and those that can't. Some of these studies seem a little silly or abstract (for example, how easy is ***parallel*** detected?), but by looking at them as whole, we can create some basic visual attribute categories that can be preattentively processed.

a

These categories are:

- **Form** (line, shape, and size)
- **Color** (hue and intensity)
- **Spatial position** (two-dimensional, stereoscopic)
- **Motion** (blink, direction)

The list of specifics within those categories can get quite long, but, thankfully, you can experiment with your graphics to find what works. If one version doesn't highlight the data, try something different. If it's easy for you to pick out what's important, chances are that it'll be easy for others. Of course, it's always a good idea to run things by others as a sanity check. Figure 6-3 shows some ways to differentiate based on preattentive attributes.

Not all preattentive attributes are created equal. Look at Figure 6-3 again. Although they all highlight the three data points, some make the three points slightly easier to see than others. In Figure 6-3(e) for example, if you used pink and red, it would be slightly more difficult to pick out the subtle difference in colors. The amount of "pop" for preattenttive attributes depends on how different the attributes are. The shapes in example in Figure 6-3(a) are more different from each other than the circles and squares in Figure 6-3(b) and slightly easier to see. It's still possible to distinguish the difference in Figure 6-3(b), but it's just not as obvious.

the

This concept of preattentive processing should be treated as just that—a concept. The line between our preattentive processing and conscious processing is blurry. When looking at a visualization, you can slip between the two quickly and quietly. With repeated exposure, you can actually train our preattentive processing. Meaning, over time, no matter how poorly designed a dashboard is, analysts will eventually pick up skills to quickly identify important features depending on environment and culture. But the point remains—if you want to direct the viewer's focus and attention, you should leverage some basic elements like form and color to highlight the point you need to make in the data.

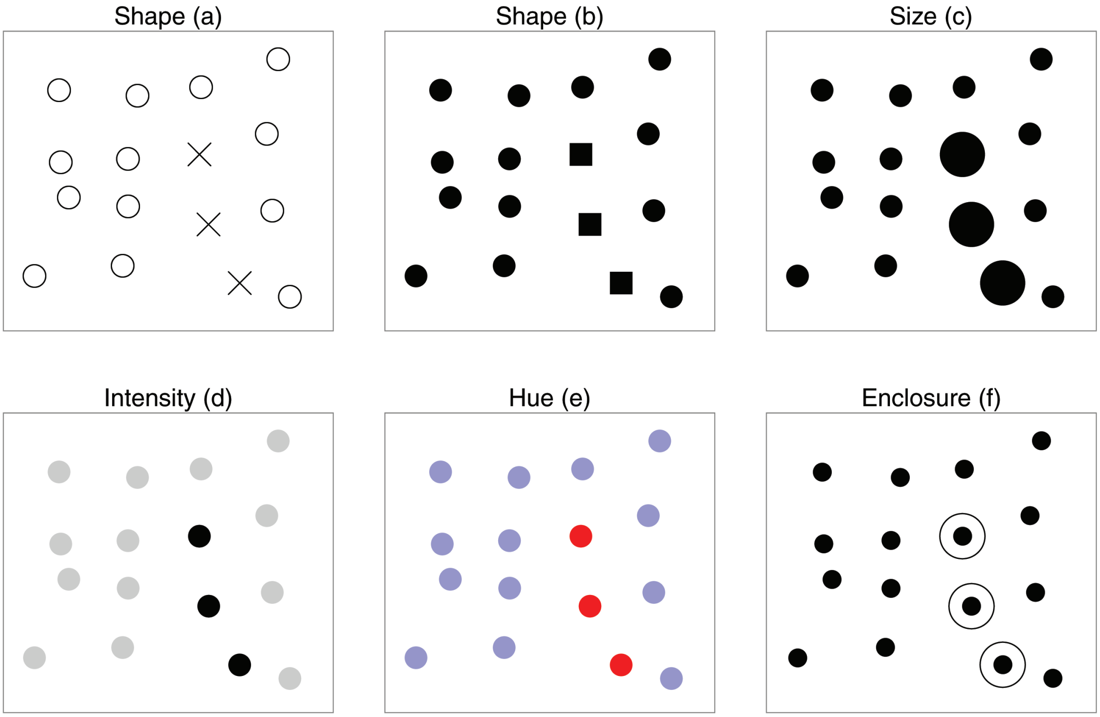


FIGURE 6-3 *Examples of preattentive attributes*

Finally, one last word of caution about preattentive processing: it's possible to overload this process and negate any benefit. Take a look at Figure 6-4.

Cap

- In Figure 6-4(a), we have separated the data into three groups and then coded them by color. It's easy to tell them apart. Not only are they spatially grouped, but color highlights their differences.
- In Figure 6-4(b), we separated the data into two groups—***different*** from the groups in Figure 6-4(a)—and then coded them by shape. It's a little harder to tell them apart, but you can still pick out the two groups.
- When we combine the methods in Figure 6-4(c), things get a bit more complicated. To separate them based on shape, you have to actively inspect individual elements and separate them consciously.

The lesson here is that you have to be careful to keep the visuals as simple as possible to exploit the viewer's preattentive processing for their benefit.

> ***Note***
>
> *This chapter has a lot of visualizations and not a lot of source code. If you are interested in how we created the figures in this chapter, the accompanying source code for the chapter is on the book's website (www.wiley.com/go/datadrivensecurity)*

monofont not italic

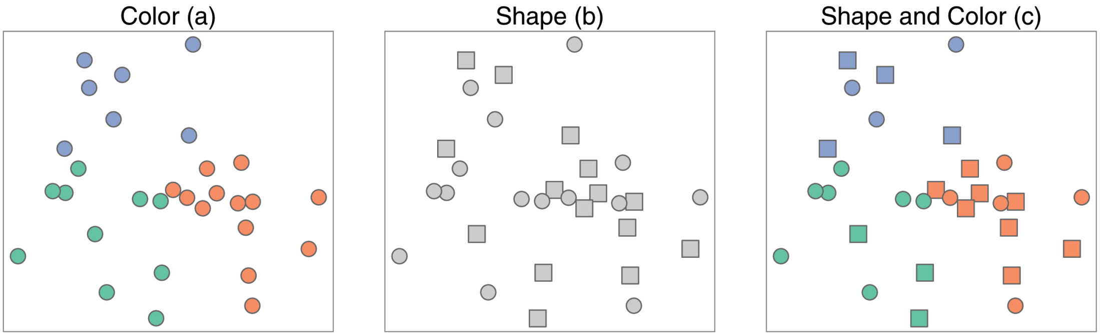


FIGURE 6-4 *Too many attributes*

Understanding the Components of Visual Communications

The chapter began by looking at how the brain visually processes information, including how you can leverage your preattentive processing and saccadic movements to increase the viewer's visual perception. This section focuses on the visual building blocks and material that you have to work with. You need to begin with the data and encode the values through various attributes like position, shape, length, and size. Perhaps you'll want to encode changes over time with slopes or angles and separate categories by color hue, saturation, or lightness. If you combine elements, you can communicate relationships and groupings. Every choice you make in creating a visualization will affect how well others will decode the data.

Avoiding the Third Dimension

First and foremost, unless you are creating a physical data sculpture, or are working with special software that allows you model in three dimensions, you are dealing in two dimensions. The screens you look at, the reports you print, and the slides you project on the wall are all limited to width and height. Of course, you can simulate a third dimension of depth, but this is a challenge. Simulating a third dimension will always be just that, a simulation.

to

In order to simulate depth, you need to change the very attributes you are using to convey the meaning of your data. Elements that are closer in the simulation will need to be bigger and those further away will be smaller. The effect from the simulated perspective will modify the viewer's ability to compare and consume the data accurately. For this reason, we strongly recommend staying away from plotting in three dimensions. Plus, two dimensions offer a tremendous amount of flexibility. Even though readily available desktop tools like Excel make 3D charts incredibly easy, you should fight the urge if your goal is to communicate your data to others.

Don't think of working with two dimensions as a limiting factor any more than just 12 notes in a chromatic scale is limiting to Western music. Much research has been conducted into communicating in two dimensions; we will highlight two seminal papers published in the mid-1980s by two statisticians—William S. Cleveland and Robert McGill. They open the first paper, "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods" with, "The subject of graphical methods for data analysis and for data presentation needs a scientific foundation." And, they did just that. They conducted

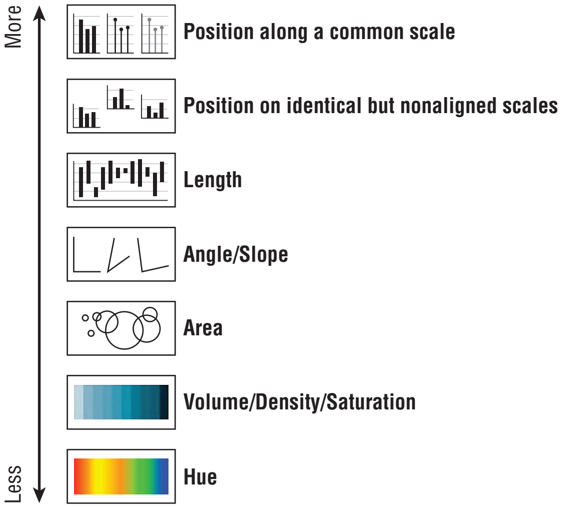

More
Position along a common scale
Position on identical but nonaligned scales
Length
Angle/Slope
Area
Volume/Density/Saturation
Hue
Less

FIGURE 6-5 *Accuracy of decoding*

experiments where subjects were shown various graphics and measured how accurately they were able to visually decode the quantitative information in them. In their second paper, "Graphical Perception and Graphical Methods for Analyzing Scientific Data," they updated their results and offered an ordered list of visual encodings and the relative accuracy in their decoding. See Figure 6-5.

These are not mutually exclusive and the distinctions between these methods can get a little blurry. For example to decode a simple bar chart, you might use position on a common scale to determine the quantity, but then use length to compare two bars within the same chart. In a pie chart, you might primarily use angles, but the area of the slice and arc length may also factor into your perception. The findings from this research should serve as a guideline. If your goal is communicating quantitative data accurately, a bar chart is always better than a pie chart and a grouped bar chart is better than a stacked bar chart.

As with all guidelines, you can deviate from this advice. Sometimes your goal is not to convey specific quantitative data, and the lack of accuracy in decoding is desired. As an example, look at Figure 6-6. When looking at the pie chart on the left, it is relatively difficult to gauge the specific difference between the five slices. Looking at just the pie chart, you'd probably conclude that they are all about equal. However, if you look at the bar chart on the right, it's relatively trivial to see the differences because you are using position on a common scale. Obviously, if you had confidence in the accuracy of the data, the bar chart on the right is far easier to interpret. But what if the data you have is from a small opinion survey? Although you can calculate precise values, the differences in the values could easily be explained with sample error. In this case, you could justify using a less accurate method to communicate the data.

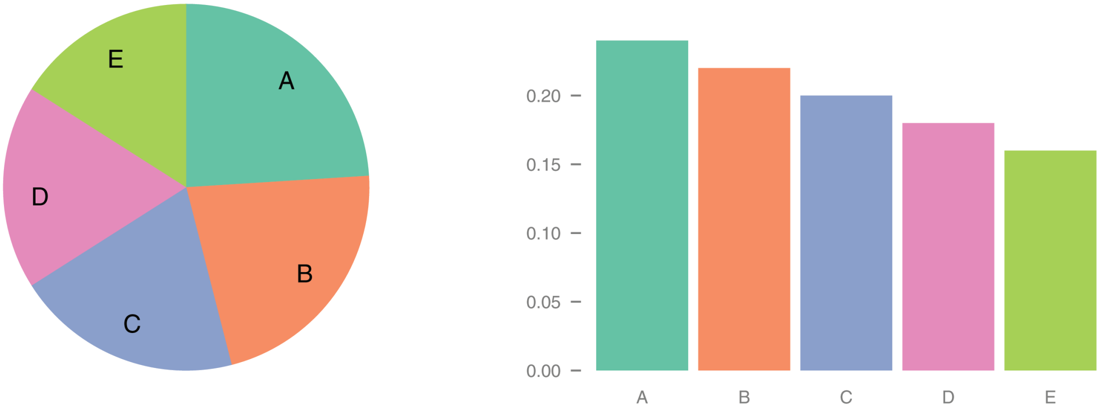


FIGURE 6-6 *Comparing pie and bar charts*

Save the Pies for Dessert

If you are new to data visualization, there are essentially two distinct (and sometimes very passionate) opinions when it comes to visualizations that use techniques lower on Cleveland's accuracy list. Pie charts are often at the center of debate since they are used (and abused) more often than others. The core argument against pie charts is that the data can always be represented better and more accurately with other methods. As Stephen Few said in his 2007 paper "Save the Pies for Dessert," "Of all the graphs that play major roles in the lexicon of quantitative communication, the pie chart is by far the least effective. Its colorful voice is often heard, but rarely understood. It mumbles when it talks." On the other side is the point we made here—that the goal of communication may not be precision. There are other less convincing arguments in the defense of pie charts, but there is one piece of common ground: choose the visualization method deliberately and be sure it communicates the message you want to send.

Au: Ref / ?

It's in the references
in the back of the book

Cap

Using Color

If you've not been tasked with selecting colors for a project, this brief introduction may make color selection seem easy. There are a few guidelines about which types of color palettes go with which types of variables and a deep well of knowledge from color research has created a handful of easy rules for palette creation. However, it won't be until you're trying yet another set of colors in your visualization that you will truly appreciate the words of Edward Tufte: "Avoiding catastrophe becomes the first principle in bringing color to information: Above all, do no harm."

comma

Tufte reference is from "Envisioning Information", should this be clarified?

Au: Ref / ?

There are many websites and tools that apply color theory to make palette selection relatively painless (see Appendix A for a complete list of resources, but ColorBrewer (http://colorbrewer2.org/) and HCL Picker (http://tristen.ca/hcl-picker/) are our favorites). With some understanding of your data, picking pertinent colors is the easy part. Colors also have to support and hopefully even highlight the message and be pleasing to the eye, which has a large element of subjectivity and is unique to each and every visual story. This creates the challenge with color: you have to balance function, aesthetics, and theory across just a handful of colors.

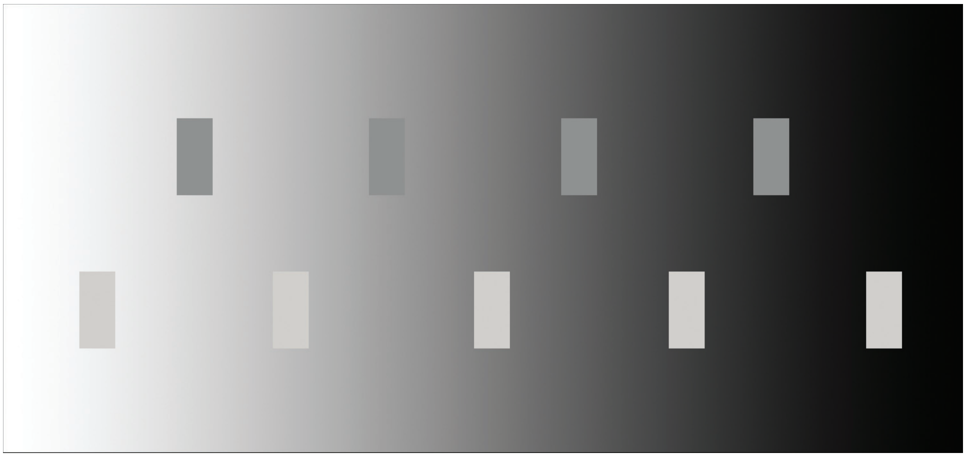
OK

[/] x2
(brackets)

Cap

Color Is Relative

The first and perhaps most important aspect of color selection is that colors are always interpreted relative to the surrounding environment. For example, Figure 6-7 shows two rows of gray boxes on a gradient background. Even if you know each row has a consistent shade of gray, you will still see different shades on the same row as you scan from side to side. And to some, the upper-left box looks to be the same color as the lower-right box. That's because you see the shade in the boxes relative to the surrounding background. The boxes appear darker on a white background and lighter on a dark background. You can use this fact to your benefit. If you want to emphasize one variable above all else, you can choose a contrasting color from the rest. For example, red shapes will stand out among shades of light blue shapes, but will blend in with pink and orange shapes.



fix alignment of figure caption

FIGURE 6-7 *Visual signal and noise detection illusion*

We Are the ~~99 Percent~~ 10 Percent

Nearly 10 percent of males and about 1 percent of the females are color blind. This means that at some point (probably sooner than you think) your visualizations and dashboards will be viewed by someone incapable of seeing the entire spectrum of the rainbow. Having some understanding of the types of color blindness can help you choose colors that everyone can see. The largest portion of color blind people have either protanomaly (red blindness) or deuteranomaly (green blindness), making red and green a poor choice to include in the same graphic. Some color-selection tools (like ColorBrewer) factor in color blindness and have an option to select colorblind safe palettes. Whatever your color tools are, keep in mind the 10 percent.

Palettes Depend on Data

OK

We briefly discussed data types in a sidebar in Chapter 3 titled, “Isn’t ‘data’ just ‘data’?” There are only a handful of high-level data types that you’ll need to be aware of and most of them fall into either categorical or quantitative values.

- **Categorical data** are represented as groups with category names, such as operating systems by type or lists of programming languages. Categorical data sometimes has a natural order. Rankings such as “first,” “second,” “third,” or “high,” “medium,” and “low” are treated like categorical values but have an added sense of order.
- **Quantitative data** are numerical values, which are things you count or measure such as bytes, packets, sessions, number of servers, and so on.

The difference between categorical and quantitative can sometimes be tricky. For example, TCP/UDP port numbers appear quantitative since they are sequential numbers going up to 65,535. But you have to treat them as categories: you would never add *echo* and two *telnet* ports to get *DNS* because the sum would make no sense in terms of port numbers. Another confusing data type is date/time. Most of the time you'll treat it as an ordered categorical variable (such as the year, month, day of week, and so on), but other times you'll store it as a quantity (seconds since the epoch) to enable calculations on time and time series data.

Cap

You have to be careful when using colors to represent a quantity. Consumers are relatively inaccurate when decoding quantity from a color scale. But color can be used in circumstances where rough comparisons are enough. For example, back in Figure 5-7 in Chapter 5, you don't need to see that exactly 1 in 724 people in Wyoming were infected with ZeroAccess. The color is simply communicating that Wyoming had more infections per person than any other state.

Figure 6-8 shows three types of color palettes—sequential, divergent, and qualitative—from the ColorBrewer website.

- You select a palette of **sequential colors** to represent quantity or perhaps ordered categorical data. Sequential color palettes are built using a single hue (blue, for example) and then adjust the lightness or saturation of that color to cover the range of the quantitative data.
- **Divergent colors** are also used on quantitative or ordered data, but help communicate above or below some middle value. Typically, the middle value is white and two divergent hues are used on either end. Divergent color scales may be used to convey two directions in the data such as above or below average (as it was used in Figure 5-7).
- Finally, you have **qualitative colors**, which are intended to simply be distinct from one another. This makes them well suited for visualizing categorical data.

Sequential

See New Graphic

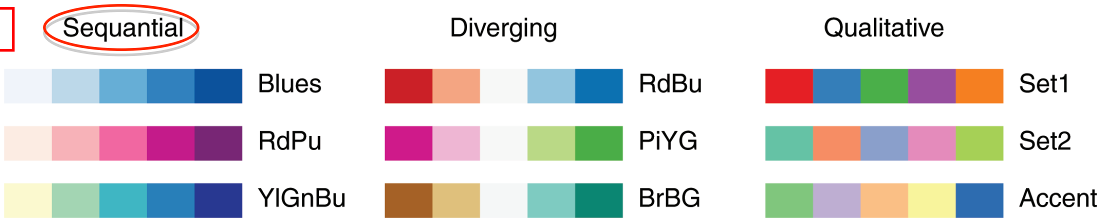


Figure 6-8 *Sample color palettes from ColorBrewer*

Putting It All Together

We've laid some good groundwork here, so it's time to look at how these things come together to help communicate your data. This section spends less time talking about how to create these and more time on ***why*** we create these as we do. All of the source data and code needed to create these visualizations in this chapter are on the book's website. Creating the basic types of plots is relatively easy using the R language and ggplot2. Most of these plots are available as options in more familiar tools such as Excel.

Using Points

The easiest method to communicate and compare two quantitative variables is the basic scatterplot. Scatterplots position points along a common scale (both x and y scales) and allow the viewer to very accurately determine the value of each variable for each data point and to make comparisons between points. It is insanely simple to create scatterplots in R (`plot(x, y)`). You can do this often just to "see" the data you are working with. For example, Figure 6-9 shows 8 hours of firewall traffic. Each dot represents total number of packets (x-axis) and total number of bytes transferred (y-axis) processed by the firewall over 5 minutes.

This is a good example of a pattern quickly jumping out of a plot. You can see that the firewall traffic for the day ranges from 7 to 19 gigabytes, and packets range from 12 to 27 million. The linear relationship is very apparent here: as you see more packets you see more bytes. Now this isn't exactly a news flash or all that informative, but a simple scatterplot can show patterns when you're not sure whether they exist. Figure 6-10 shows an example of a scatterplot that reveals something you didn't know. The time of day is along the x-axis and the number of sessions is on the y-axis.

Cap / comma

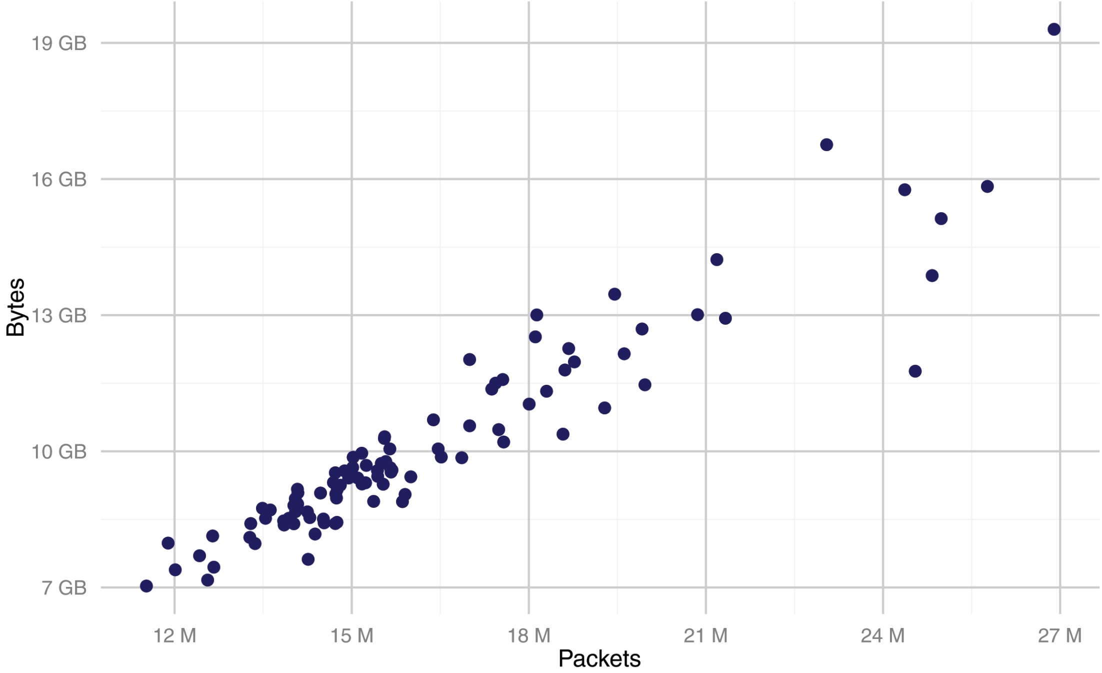


Figure 6-9 *Basic scatterplot*

Yes, change this to say "The scatterplot in Figure 6-10 has a few…"

(Figure 6-10) / ?

This scatterplot has a few extra features. Note the faint lines down from the points; they give just a hint of a bar chart and visually tie the points (which are rather bunched up) back to the x-axis. To highlight the repeating element of time, the line at the top of the hour is thickened (and falls on the grid lines); the points also change to red every 30 minutes. There is a noticeable dip at the top of the hour and not much change at the half-hour marks, and it's important to emphasize those times for easier comparison (remember the preattentive processing?). What is the cause of this dip? Perhaps this organization has a meeting-heavy culture and network activity drops as people head to their next meeting? You can't know the cause from this data, but the dip pattern really jumps out with a simple scatterplot.

comma

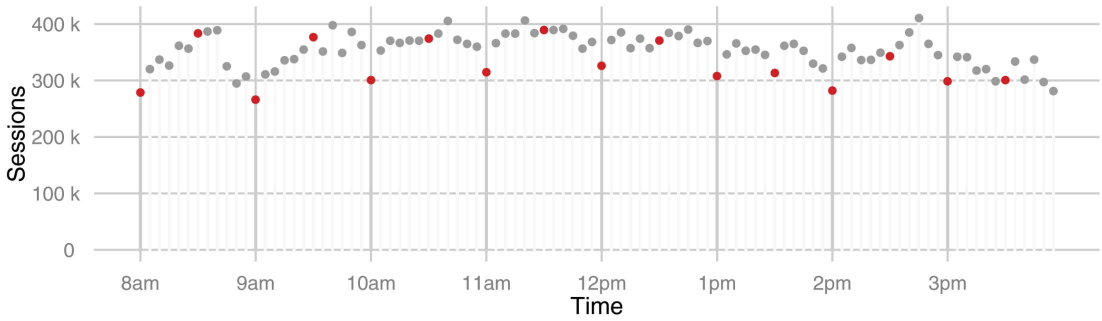


FIGURE 6-10 *Dot plot: packets over time*

Creating Directions with Lines

You may have heard at some point that "lines are just points in motion," and that's true—you generally see lines having a sense of direction. In this section, you'll take the same firewall traffic and separate the types of devices on the network:

- Desktops
- Servers
- Printers
- Networking equipment

You'll see two plots: first, the same type of scatterplot as the time series and then a line plot (see Figure 6-11).

comma

Comp: Don't split para

It's rather clear what's going on with the line plot and it's easy to follow the traffic over time for each of the four devices. The scatterplot on the left is a little difficult to follow, although you can see trends and

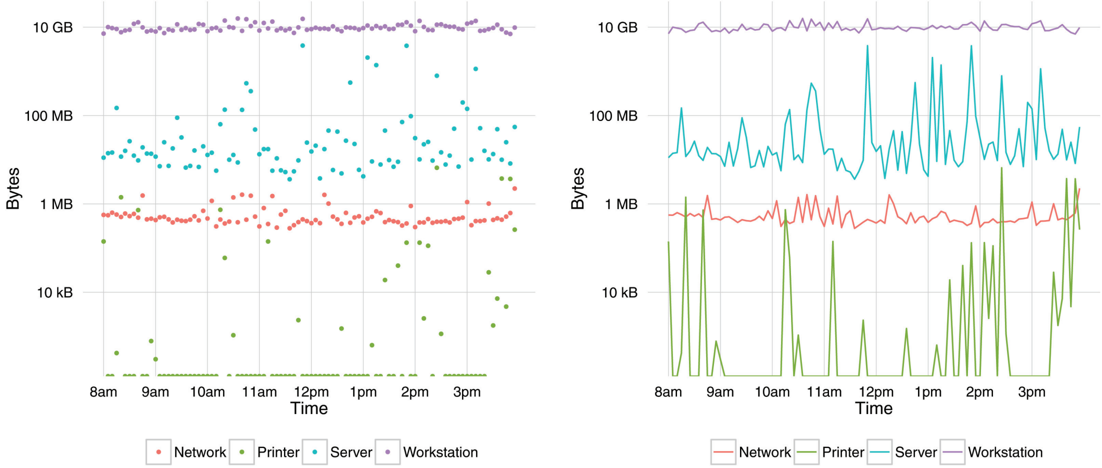


FIGURE 6-11 *Line plot: traffic by device*

differences between the categories. Line plots are quite good at accurately communicating data; they compare points on the line along a common scale and use the slope of the line as a sign of change. For example, notice the steep slopes in the data series for printers. Most commonly, line plots have an ordered variable on the horizontal axis (often "time") and one or more quantitative variables on the vertical axis. (It's possible to flip the orientation depending on the presentation circumstances.) In this case, you are plotting number of packets (quantitative) on the y-axis against successive five-minute periods (ordered) with each line representing a category of device.

Log Scales for Logs

In Figure 6-11 the y-axis is plotted on a logarithmic scale. Notice how the values on the axis increase by powers of ten for a given physical distance on the plot. If this plot was plotted on a linear scale, you'd see the workstation traffic at the top and the other three lines would be reduced to almost zero. We chose a log scale because we needed to show these data series on the same chart, even though they differed by three orders of magnitude. . You have to be careful when you use a log scale. Most people in business are used to seeing linear scales, and they are conditioned to do comparisons assuming a linear scale. For example, the viewers might come to the conclusion that the networking equipment has about half the traffic of the workstations because it's visually about half the distance from the axis. But, in reality, workstations are generating about 10,000 times more traffic than network devices. If the logarithmic scale isn't clear to the viewers, they could draw incorrect conclusions.

Building Bar Charts

Bar charts are one of the most effective ways to communicate when one variable is quantitative and the other variable is categorical. There are a few variations on the basic bar chart. Figure 6-12 shows three different ways of displaying vulnerability counts and severity classification per device. On the far left, you have a typical bar chart with vertical bars. One simple modification (not shown) would be to flip the orientation so that the bars are horizontal. The difference between vertical and horizontal orientation is largely aesthetic and depends on where the chart will appear. The vertical bar chart is simple: the length of each bar is proportional to the total number of vulnerabilities for each device type. You can easily see that workstations have the most vulnerabilities, and servers are close, with 20 percent less or so. In comparison, it is obvious that the number of vulnerabilities in networking devices and printers are quite small.

Cap

The other two bar charts have an additional categorical variable for the severity of the vulnerability —High, Medium, or Low. The stacked and group bar charts use sequential color scheme severity levels. With the stacked bar charts, you are still able to compare totals. It's still clear that workstations have more vulnerabilities than all others. But comparing across severity is difficult as you lose the common scale. As an example, attempt to visually compare the high vulnerabilities of workstations to servers. Since they are not aligned you judge them purely by length on a non-aligned scale and are therefore less accurate.

Now look at the grouped bar chart and it quickly becomes clear that workstations have more high-severity vulnerabilities than servers. The one drawback to the grouped bar chart is that you lose the overall count comparison. When the overall totals are close, it's more difficult to tell that workstations have more vulnerabilities overall from the grouped bar chart. The type of bar chart you choose is largely dependent on the message you are trying to send.

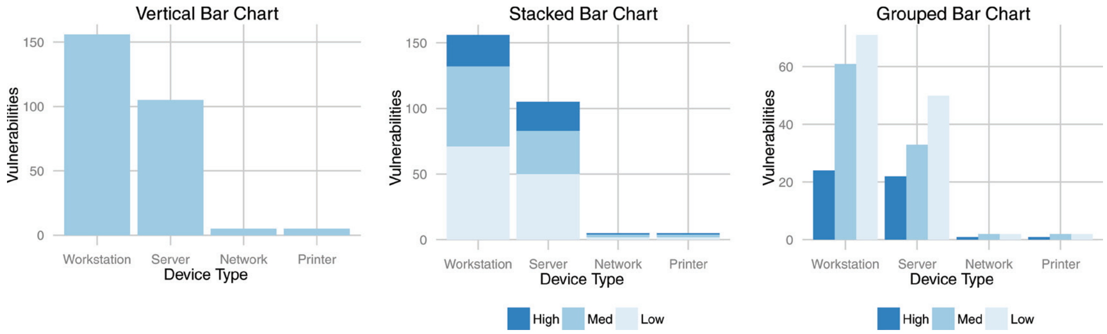


FIGURE 6-12 *Bar charts: vulnerability counts*

Au: colors inFig.6-12 as wanted?

Leveraging Opacity

Another technique for communicating differences in variable values is the opacity or transparency of colors within graphs. If the data is overlapping or dense and you plot it with a solid opaque color, you have no way of knowing just how many points are stacked up underneath that. Luckily, you can simply make the color semi-transparent. This will allow any points beneath to show through a given point. Within R there are two methods for doing this. First within `ggplot2` most (perhaps all) of the chart types allow for an alpha setting between 0 and 1 (***alpha*** is the term used in color specifications to define opacity). Or you can code the alpha right into the color with a 4th byte, meaning a red value of *#FF0000* is the same as *#FF0000FF* (with the last *FF* setting opacity to maximum). If you want to set opacity to 50 percent, 255/2 = 128 = 0x80, so you can set the color to *#FF000080*. The red color is now 50 percent opaque. The benefit of adjusting the alpha (opacity) is demonstrated in Figure 6-13.

Fig 6-13 HERE ->

These two charts show the same data: 8 hours of firewall data for networking devices split into 5-minute totals. The number of network sessions is plotted along the x-axis and the number of bytes is on the y-axis. The size of each point ("bubble") is proportional to the packet count. One challenge in this visualization is that many points overlap. By setting the alpha value to 1/3 in the chart on the right, you can see through any bubble to bubbles that lie underneath it.

It's handy for you to set the alpha as a fraction (such as 1/3 instead of 0.33) because it makes it obvious how many points or bubbles will stack up to equal the maximum color value (solid). This allows you to tweak the alpha for how many layers you have. If you have 50 layers (some of the maps in Chapter 5 have code that use small alpha values like this), you can set the alpha to 1/50 (as opposed to converting to 0.02 and typing that in).

OK

Size Encoding

Figure 6-13 is encoding another quantitative variable by mapping the size (***area***) of the circle ("bubble") to the number of packets in a 5-minute period. Looking back at the accuracy chart in Figure 6-5, you can see that ***area*** is relatively low on the list. This difficulty is compounded in Figure 6-13 because there is no legend for bubble size (an intentional oversight on our part). But for the purposes of this chapter, note the relative values here—the relatively large versus relatively small number of packets. In a more formal setting,

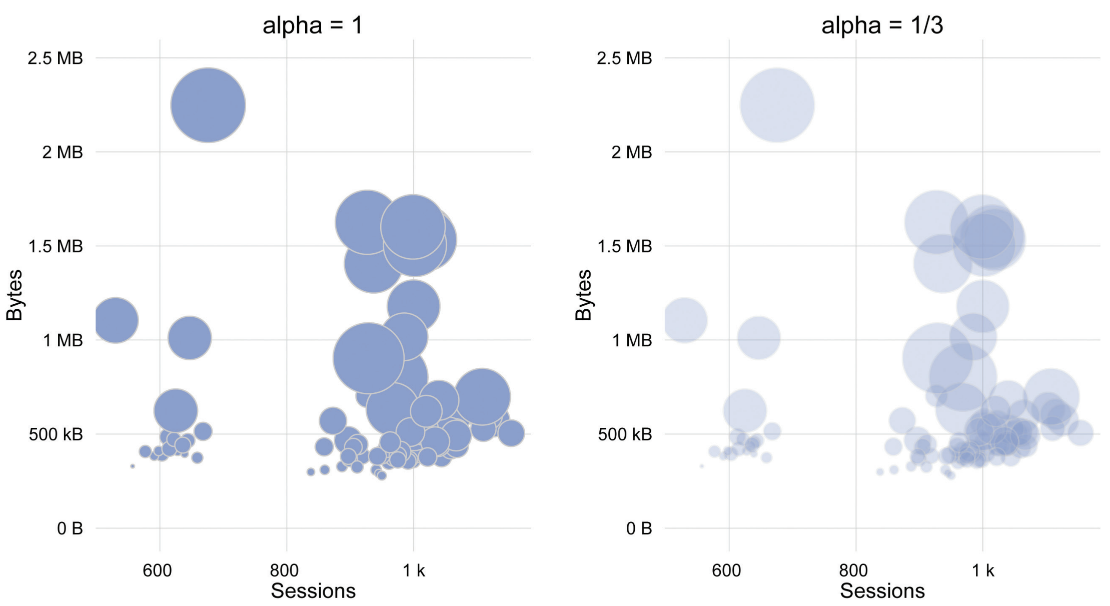


FIGURE 6-13 *Bubble chart: opacity shows stacking*

Comp: Move Fig.6-13. Don't split para.

No, please move to previous section where it is first referenced.

you'd want to add a description to the title or use some other annotation to indicate the significance of the "bubble" size. For this purposes of this exercise, you are simply looking for any obvious patterns and this type of graphic shows relative sizes. Bubble charts like this are good for crude estimates, and are thus similar in communication capability to pie charts.

the

Another visualization method with similar traits is the ***treemap***. A treemap uses ***area*** and ***color*** to encode two ***quantitative*** variables (see Figure 6-14). The treemap visualization method relies on area, and thus is relatively low in visual accuracy.

Figure 6-14 somewhere in this section

Figure 6-14 portrays the number of devices on a network and the traffic volume for each type. It uses the size of rectangles to communicate a quantity and the color of the rectangle to communicate a different quantitative variable. Often times the rectangles are visually grouped to depict categorical relationships. In Figure 6-14. the size of each rectangle is proportional to the quantity of devices on the network by type (workstations, servers, and networking devices) and, the lightness of color of each rectangle is proportional to the volume of traffic they produce (normalized).

tr

We should reiterate—a treemap combines two relatively inaccurate methods of encoding quantities. This makes treemaps difficult to execute well and often confusing to viewers. The same rule applies to treemaps as to pie charts and bubble plots—there are usually better visualization methods to communicate the data.

Communicating Distributions

Sometimes you'll just want to show the values within single variable and how they are distributed. Within classical statistics, you have descriptive statistics that attempt to reduce a distribution to a set

a

of descriptive values. For example, if you go back to the 8 hours of firewall data shown in Figures 6-9 and 6-10, you could describe the distribution of total sessions within each 5-minute window like this:

Description	Statistic
Min	265,800
Median	356,500
Mean	350,500
Standard Dev.	32,093
Max	410,700
Skew	–0.5
Kurtosis	–0.457

reduce width of table

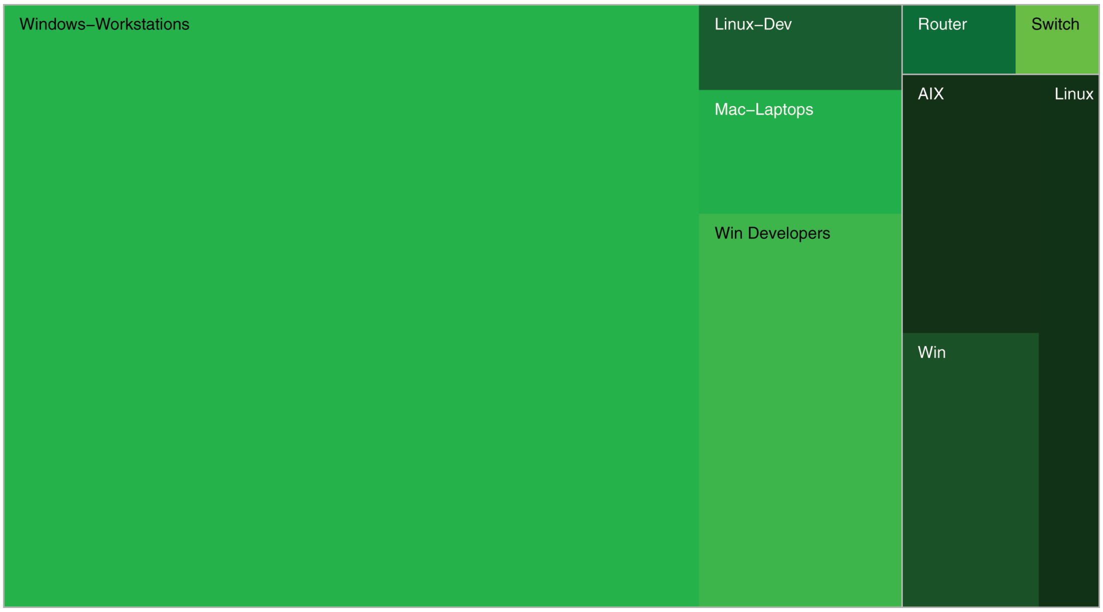


FIGURE 6-14 *Treemap: devices and traffic on the network*

Move Fig.6-14 beneath para?

Move up. This is referenced in previous section on Size Encoding, not on distributions

Most people can't look at these numbers and understand what the data is telling them. Nor will they be able to see any subtle patterns; descriptive statistics are about reducing a distribution of values to a set of individual numbers. This is where visualizations can help out considerably.

Histograms and Density Plots

Rather than reduce a distribution to a few descriptive statistics, you can represent every value in the variable. Figure 6-15 shows a basic histogram on the left and a density plot on the right, both for the same data set.

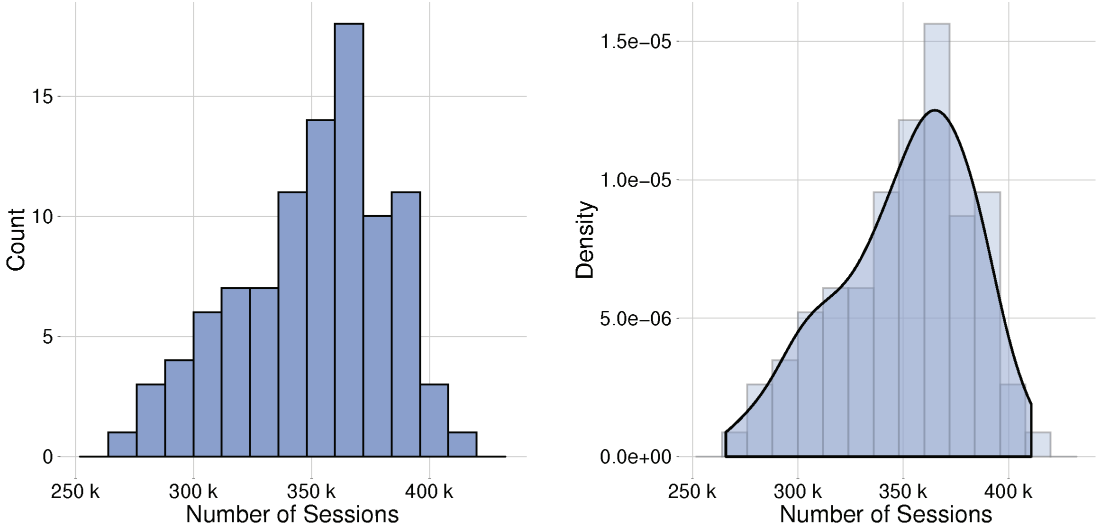


Figure 6-15 *Histogram and density plot of the firewall sessions*

A histogram uses a simple process called ***binning***. It works by creating equally spaced "bins" and then counting how many of the measurements are in each bin. In this example, we created bins that are 12,000 sessions wide. You can see that at the peak—around 350,000 sessions—we had about 18 sessions within that bin. Part of the criticism of histograms is that you can affect how histograms appear by adjusting the size and position of the bins. But these plots are indispensable when you want to get a feel for a distribution and they are quite effective in communicating the basic shape.

The plot on the right in Figure 6-15 is a density plot. It uses the same approach as the histogram, but the bins are quite small and a smoothing process is applied over it. By projecting the original histogram behind it, you can see how it flattens the peaks and diminishes the valleys. There's no right or wrong between the two—both involve some approximations. When you are exploring your data, it's quite easy to pass in data to `hist()` and get an immediate (although maybe not pretty) histogram.

Boxing in Boxplots

OK

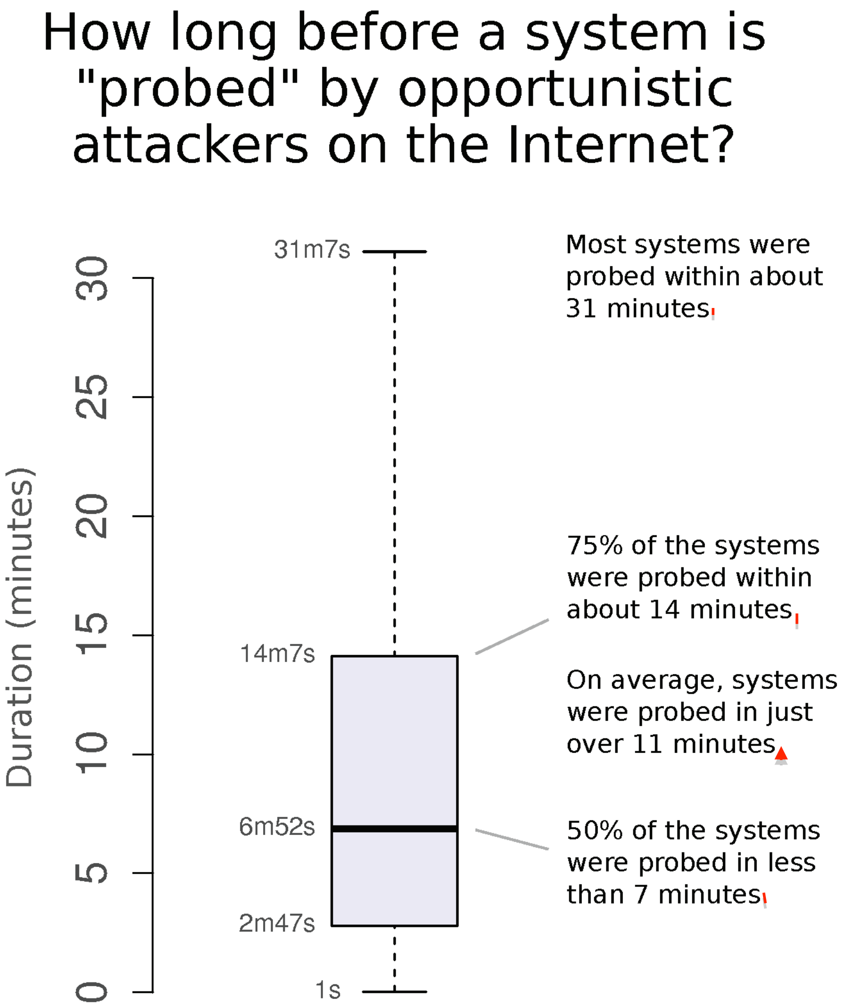
Another method, which was developed by John Tukey (remember him from Chapter 1?), is the boxplot, which we touched on in Chapter 5 when discussing outliers. This is not something your viewers will intuitively understand if they haven't seen one before, so it may require a little more supporting material than other methods need. In the fall of 2012, one of the authors (Jay) set up a simple honey pot to record the packets it saw on the Internet. How often is a host scanned when it's on the Internet? You can get a feel for the answer in the boxplot in Figure 6-16.

OK

Author: Any chance you can reduce the font size in the figure and resupply?

This does stand out, but this is an old graphic I created and I don't have the source any longer. Could the graphics folks work with the PDF as a vector graphic and improve it?

What happens if this is just scaled down?



period x4

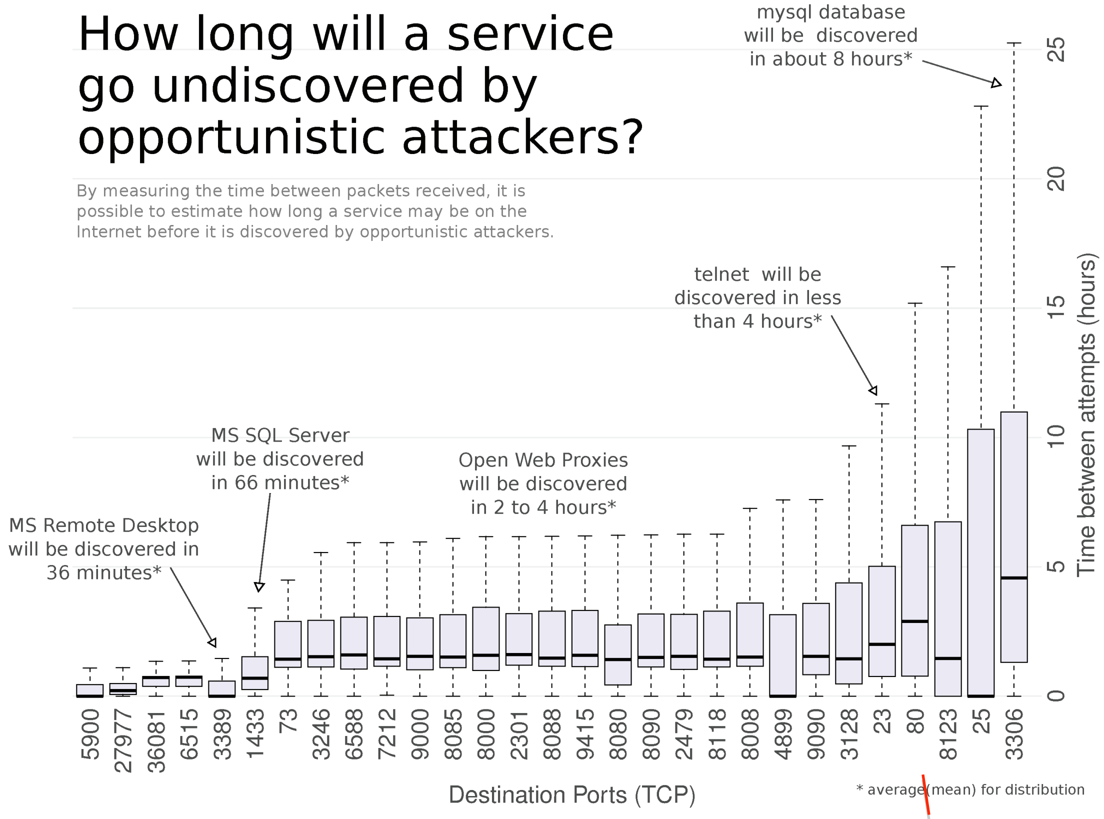
FIGURE 6-16 *Honey pot traffic: boxplot*

The boxplot begins with the median (middle) value of the distribution and it places the center bar there. Then it computes the 25th and 75th percentiles. This means that 25 percent of the data is below the 25th percentile, 25 percent of the data is above the 75th percentile, and 50 percent of the data is between the two. These two points form the length of the box and represent the ***inter-quartile range*** or IQR.

There are a few different methods used to represent the length of the lines. The most common method places the lines one and a half times the IQR away from the box. Other methods place the end of the line at the minimum and maximum of the data. Figure 6-17 attempts to convey a large number of distributions within one chart with boxplots.

Au: are you happy w/ the large font in the heading?

If the graphics folks could reduce it, I think it would look better with a smaller font.



space

FIGURE 6-17 *Boxplots: opportunistic packets*

What's interesting about Figure 6-17 is that it was generated with over 100 million values. It not only conveys a large quantity of data, but it's also able to represent a certain amount of confidence in the data. In this case, just stating the mean or median would have been a disservice, since some of these have a very wide range of possible observations. How well could you have explained these values and the variations with anything other than a visualization of the distributions?

Visualizing Time Series

This chapter has glossed over time series data even though you've been working with it in most of the visualizations. Time series data are data collected over the same and repeated time intervals. For most of the firewall figures in this chapter, we parsed the log files and counted up the bytes, sessions, and packets within a sequence of 5-minute time windows. This allows aggregation of individual entries into more manageable data points. But depending on how you slice up time and aggregate the data, you can get and see different types of things.

Figure 6-18 is looking at 21 days of firewall traffic sliced into 5-minute chunks. This is quite a bit of data for a small line graph (over 6,000 data points in a few inches), and when you try to represent that data with a line plot, the lines crisscross over one another so much that they look like one thick and jittery line. If you try to reduce the mess by simplifying the underlying data with an hourly average (the middle plot in Figure 6-18), you lose the extremes and the details, which is not generally good in a field like information security where extremes matter. In the bottom plot, we replaced the lines in the first plot with points. This removes much of the mess and allows you to see the general trends and the extreme points.

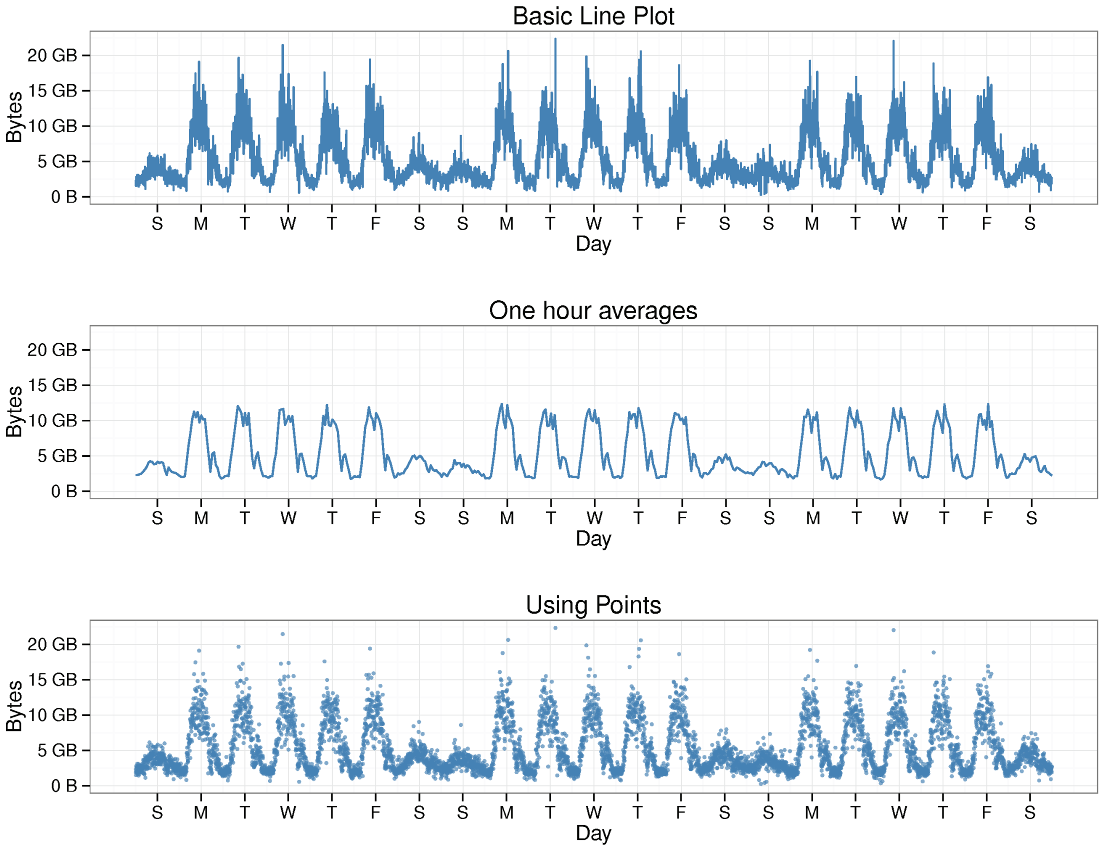


FIGURE 6-18 *Time series: 21 days of traffic*

Time series data can get very dense to visualize when you are talking about data from logs. We even made it easier by looking at 5-minute slices instead of 1-minute slices. How you prepare and visualize the data is dependent on what you are looking for in the data. If you are looking for specific spikes or gaps in traffic, you should avoid using a rolling average. However, if you want to understand general patterns, averages are usually good enough.

Experiment on Your Own

We've covered quite a few techniques so far in this chapter. Feel free to get creative and try one or more techniques on your time series data. What if you tried showing each hour with a boxplot? What if you used larger points and varied color based on size and turned down the alpha? Creating good visualizations is generally an iterative process, so take this as a license to experiment! Remember that you aren't limited to static visualizations. You can create interactive visualizations (as you'll see Chapter 11) or turn the time series into a video fit for YouTube, as you will see in the next section.

Turning Your Data into a Movie Star

This chapter has focused primarily on foundational components of data visualizations. These apply to static or interactive graphics, dashboards, and as you'll now see, to videos as well. One of the more fun "tricks" we've learned is how to turn data into a video. In order to do this, you combine two techniques: automated sequential graphics and stop-motion software.

If you aren't familiar with stop-motion by name, you're certainly familiar with it by sight. It's the Claymation technology of setting up a scene, taking a picture, and then changing it slightly, taking another picture, and so on. When you string all of those pictures together you get the appearance of motion and you have a video. Same concept with data animation, but instead of taking a picture, you want to generate a graphic and save it as a picture. Then you can use any number of stop-motion software packages (mencoder, avconv, FFmpeg, iMovie, and so on) to create a movie from the pictures. If you want to get fancy, you can include music or voice-overs so you can explain the data as it's progressing.

To see how this looks, try Listing 6-1 in an open R session.

The 6-0 listing is for setting directories and libraries. This is just an example not using data or libraries. 6-1 is okay

Au: Should this be Listing 6-0 / ?

Au: Is color correct?

"Listing 6-1" should not appear twice, so color on the second one doesn't matter, should be removed.

LISTING 6-1

```
Listing 6-1
# random walk
set.seed(1)
# set up nine directions
dirs <- matrix(c(rep(seq(-1, 1), 3),
rep(seq(-1, 1), each=3)), ncol=2, byrow=T)
# start in the center
cpos <- matrix(c(0, 0), ncol=2)
# set full screen
par(mar=c(0,0,0,0))
# take 200 steps
for(i in seq(200)) {
  plot(cpos, type="p", col="gray80", xlim=c(-20, 20), ylim=c(-20,20),
yaxt="n", ann=FALSE, xaxt="n", bty="n")
  cpos <- rbind(cpos, cpos[nrow(cpos), ] + dirs[sample(1:9, 1), ])
  points(cpos[nrow(cpos), 1], cpos[nrow(cpos), 2],
type="p", pch=16, col="red")
  Sys.sleep(0.1)
}
# reset screen back to default
par(mar=c(5.1,4.1,4.1,2.1))
```

This code will set up a matrix of nine directions. It will loop 200 times, adjusting the point in some random direction and drawing the new plot for it. It then will sleep for a tenth of a second so you can view the plot. On all but the slowest machines this looks like a random walking point on the screen. If you want a challenge, modify this script to write each of the images (hint, take a look at `help(png)`) and then create a video of it. We've done this and it's available at `http://datadrivensecurity.info/book/ch06/movie/chapter6-movie.mov` (or as part of the Chapter 6 materials on the book's website at `www.wiley.com/go/datadrivensecurity`) if you'd like to see this random walk in action!

rebreak

Summary

Communicating data visually allows you to communicate complex data and relationships quickly, enable pattern recognition, spot anomalies, and gain new perspectives. Understanding how the brain processes and stores information allows you to create visuals that leverage preattentive cues and minimize saccadic movements for efficiency. Through the work of Cleveland and McGill, you've learned that some visual methods are better for communicating quantity (in contrast, some methods "mumble"). You should combine these lessons into colors, points, lines, and shapes to communicate the stories you uncover in the data.

Recommended Readings

The following are some recommended readings that can further your understanding on some of the topics we touch on in this chapter. For full information on these recommendations and for the sources we cite in the chapter, please see Appendix B. Resources for visualization are plentiful and we had a hard time keeping this list as short as we did.

OK

Data Points*: *Visualization That Means Something* by Nathan Yau**—This is Nathan Yau's second book on visualization, and it offers a gentle introduction to the topic of visualization. His first book, ***Visualize This, offers examples and source code if you'd like that more. But both books are good places to start exploring visualizations.

***Show Me the Numbers*: *Designing Tables and Graphs to Enlighten* by Stephen Few**—Stephen Few is known for his many technical contributions to the visualization field. His books lean towards the technical side, yet carry coherent and valuable lessons for communicating through data visualizations.

toward

***Envisioning Information* by Edward R. Tufte**—Another well-known name in the field, but with a much more design-centric approach with a little more emphasis on aesthetics and function. Any one of Tufte's books is worth their price, and if you can catch his touring seminar, his books are usually included in the price of the registration.

***Information Visualization* by Colin Ware**—This is a hardcore book on the mechanics and cognitive science behind visualization. If you are really curious how humans interpret data visually, this is the book to read.