

The Perceptual and Cognitive Limits of Multivariate Data Visualization

Stephen Few, Perceptual Edge

September 2019

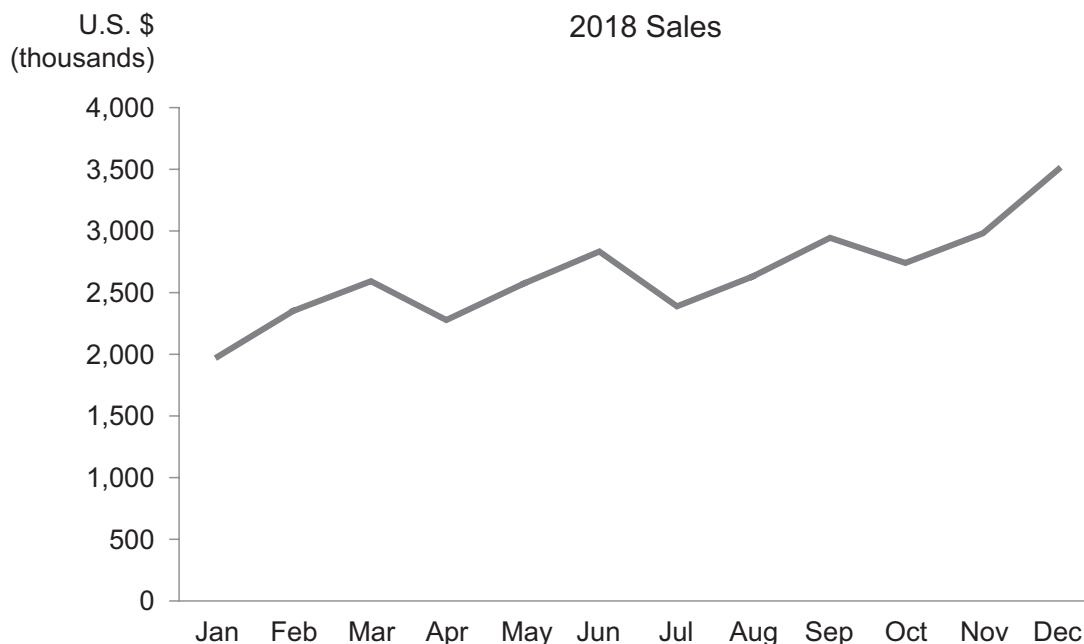
Almost all data visualizations are multivariate (i.e., they display more than one variable), but there are practical limits to the number of variables that a single graph can display. These limits vary depending on the approach that's used. Three graphical approaches are currently available for displaying multiple variables:

1. Encode each variable using a different visual attribute
2. Encode every variable using the same visual attribute
3. Increase the number of variables using small multiples

In this article, we'll consider each.

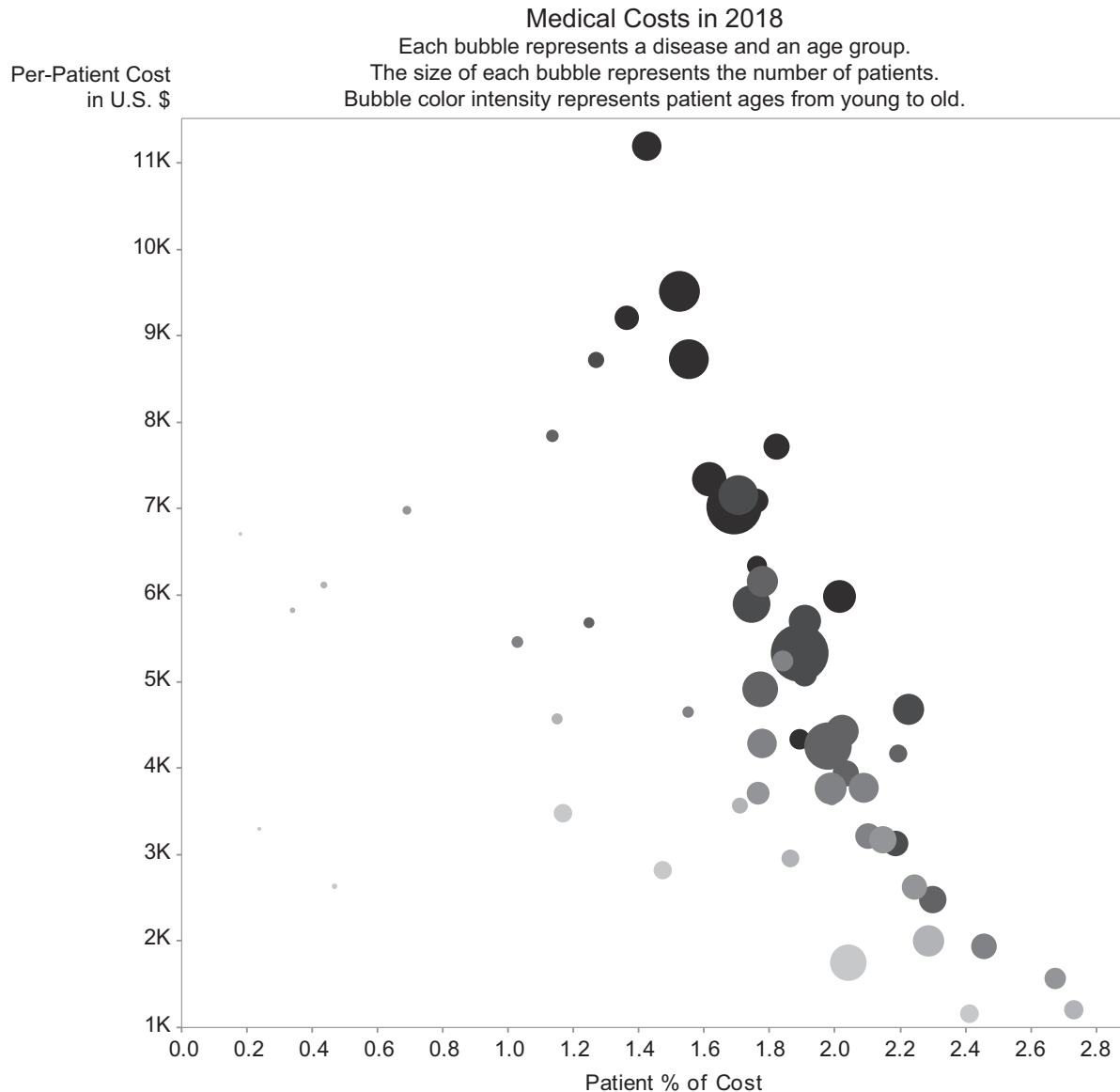
Encode Each Variable Using a Different Visual Attribute

This first approach is the most common, and it works quite well, but it typically limits the number of variables that can be effectively displayed in a single graph to four. Here's a simple example of this approach that displays only two variables:



One variable—time by month—is encoded as horizontal positions along the X axis and the other variable—sales in dollars—is encoded as vertical positions along the Y axis. In other words, this example uses two visual attributes to encode values, one per variable: 2-D horizontal position and 2-D vertical position.

Here's another example, but this time four variables are on display:



The following four visual attributes have been used to encode the four variables: horizontal position along the X axis (patient percentage of cost), vertical position along the Y axis (per patient cost in U.S. dollars), bubble size (number of patients), and bubble color intensity (patient age). Could we include a fifth variable in this graph in a way that works for our brains? There are certainly several more visual attributes from which to choose, but would any of them work in this case? Unfortunately, due mostly to perceptual limitations, the answer is, "Not well."

If you doubt this, use your imagination to consider the possibilities. Perhaps it occurred to you that, in addition to variation in color intensity, which in this case encodes patient ages, we could encode a new variable, such as patient racial group, using various hues of color. If we did this, color intensity would no longer work effectively because it is difficult to compare the varying intensities of different hues, and the variable that's encoded using hue would suffer because it is no longer easy to group objects with the same hue when color intensity varies.

As an alternative, perhaps the bubbles, which are all circular in shape, could vary in shape to encode a fifth variable (e.g., circles, squares, triangles, etc.). The problem with this approach is that, whereas we can roughly compare the sizes of circles to one another or squares to one another or triangles to one another, we cannot do a good job of comparing the sizes of circles, squares, and triangles to one another. Differences in shape make differences in size difficult to discern.

Even with only four variables, we're already pushing the limits of effectiveness in this graph. Notice how difficult it is to determine the color intensities of small bubbles and to compare them to other bubbles. Colors become difficult to discriminate when objects are tiny. The larger the object, the more color there is, which makes discrimination easier. As you continue to consider other visual attributes that might be used to encode a fifth variable, you'll encounter problems with each.

You might be thinking that I'm ignoring a visual attribute that could easily be added to this bubble plot: positions along the Z axis. Actually, I'm avoiding the Z axis for a good reason. Turning this into a 3-D graph by adding a Z axis would make the variable that's encoded along that axis incredibly difficult to read. This is because, contrary to the ease with which human perception discerns differences in 2-D position (either horizontal or vertical along a flat plane), our perception of depth is not very good. Adding a Z axis would force us to constantly rotate and tilt the graph to reorient the Z axis either horizontally or vertically in an effort to see where bubbles fall along the axis, which isn't practical.

Visual perception and cognition impose firm limits on the number of variables that we can encode in a single graph when we're using a different visual attribute for each. These limitations are tied to several factors:

1. Only a few visual attributes work well for encoding data in graphs.
2. Using some visual attributes eliminates the possibility of using certain other attributes in the same graph.
3. Working memory can only attend to three or at most four chunks of information at a time, so limited value is added by including more than four.
4. Increasing the number of visual attributes in a single graph beyond a certain number creates a cluttered appearance that undermines perception.

Let's consider each of these limitations in turn.

Effective Visual Attributes

Beginning with the work of Jacques Bertin, author of *Sémiologie Graphique* (*The Semiology of Graphics*), in the 1960s, people have studied visual perception as it applies to data visualization. Bertin explored the opportunities and limitations that influence the use of various visual attributes for encoding data. Since Bertin's seminal work, the best books on this topic have been written by Colin Ware: *Information Visualization: Perception for Design* and *Visual Thinking for Design*. Everyone working in the field of data visualization should read these books. Vendors developing data visualization products should definitely read these books, but it seems that, based on the ineffective features that most products exhibit, they rarely do.

All data visualizations have one thing in common: they encode data values graphically, using basic attributes of visual perception. Whenever we look at an object in the world, the visual representation that appears in our heads is constructed from a small set of basic visual attributes. These attributes are called *preattentive attributes of visual perception*, for they are processed in the visual cortex of the brain preattentively (i.e., prior to conscious awareness). Each of these attributes is perceived separately, but in parallel rather than serially, more rapidly than conscious perception. The speed and ease of preattentive perception is a big part of the reason why data visualization is so powerful when done properly.

Here's a fairly comprehensive list of the preattentive attributes of visual perception that are potential candidates for encoding data in graphs, grouped into six categories:

Attributes of Position

- 2-D horizontal position (i.e., objects arranged along an X axis)
- 2-D vertical position (i.e., objects arranged along a Y axis)
- Stereoscopic depth (i.e., perception of the distances of objects from the viewer, which can be simulated graphically by arranging them along a Z axis)

Attributes of Size

- Line length (e.g., the length of a bar in a bar graph)
- Line width (e.g., the width of a line in a line graph)
- Area (i.e., the 2-D size of an object, such as the size of a circle)
- Volume (i.e., the 3-D size of an object, such as the size of a sphere)

Attributes of Form

- Line orientation (e.g., the slope of a line in a line graph)
- Simple shape (e.g., differences between circles, squares, and triangles)
- Angle (i.e., the angle created where two lines meet, such as the angles formed by slices in a pie chart at its center)
- Curvature (e.g., the degree to which a line is curved)

Attributes of Appearance

- Hue (e.g., red, green, blue, etc.)
- Color intensity (i.e., the degree to which the color of an object varies from light to dark, pale to saturated, or both)
- Transparency (i.e., the degree to which we can see through an object)
- Blur (i.e., the degree to which an object appears sharp or fuzzy along its edges)
- Texture (i.e., various patterns on the surface of an object such as the grain of wood or the smooth appearance of metal)

Attributes of Movement or Change

- Direction of motion (e.g., the direction in which bubbles move in an animated bubble plot)
- Speed of motion (e.g., varying speeds in the movement of bubbles in an animated bubble plot)
- Speed of flicker (i.e., the speed at which an object flickers on and off or from low to high intensity)

Attributes of Quantity

- Numerosity (i.e., our ability to recognize differences in quantity between one, two, or three objects)
- Added marks (i.e., the varying addition of another component to an object—it is either there or it isn't—such as a border around a bubble in a bubble plot)

We can consider all 21 of these preattentive attributes of visual perception as candidates for encoding values in graphs, but only a few of them work well.

We perceive some preattentive visual attributes quantitatively. By this, I mean that we naturally perceive different expressions of the attribute as representing either greater or lesser values. For example, we perceive a long line as greater in value than a short line or a dark circle as greater in value than a light circle. We perceive each of the following attributes quantitatively:

- 2-D horizontal position (right is greater than left)
- 2-D vertical position (high is greater than low)
- Stereoscopic depth (farther away is greater than near, or vice versa)
- Line length (long is greater than short)
- Line width (thick is greater than thin)

- Area (large is greater than small)
- Volume (large is greater than small)
- Line orientation (steep is greater than shallow relative to a horizontal baseline)
- Angle (wide is greater than narrow)
- Curvature (curvy is greater than straight, or vice versa)
- Color intensity (dark or bright is greater than light or pale)
- Transparency (opaque is greater than transparent)
- Blur (fuzzy is greater than sharp, or vice versa)
- Speed of motion (fast is greater than slow)
- Speed of flicker (fast is greater than slow)
- Numerosity (more is greater than fewer)

Of these 16 attributes, only three work well for encoding quantitative data in graphs:

- 2-D horizontal position
- 2-D vertical position
- Line length

When I say that they work well, I mean that they can be perceived and compared to one another quickly, easily, and with a great deal of precision. Whereas these three attributes work well, all of the others provide only an approximate sense of value and a rough means of comparison. Of these, the following two tend to be most useful in graphs:

- Color intensity
- Area

Because color intensity and area only support approximate decoding and rough comparisons, however, we should only use them when neither 2-D horizontal position, nor 2-D vertical position, nor line length are available.

It doesn't usually make sense to even consider numerosity because it's severely limited. Numerosity refers to our preattentive ability to see differences between quantities of one, two, or three. We can also discern that more than three objects are greater than three, but we cannot decode the actual number preattentively. For example, if several clusters of dots appeared on a screen, we could recognize without conscious effort that some contained one dot, some two, some three, and some more than three. When clusters contained more than three dots, however, we could not tell how many there were without taking time to consciously count them. As such, numerosity is only useful for encoding values in a graph if quantities don't exceed three. This situation happens too rarely to routinely consider numerosity as a candidate for encoding values in graphs.

The remaining quantitatively perceived attributes—stereoscopic depth, volume, line orientation, angle, curvature, blur, speed of motion, and speed of flicker—are rarely used in graphs, either because we perceive them less well than others or because they aren't practical.

Some visual attributes can only be used to encode categorical variables, not quantitative variables. These include the following:

- Simple shape
- Hue
- Texture
- Added marks

The two that are most useful in graphs are hue and simple shape, in that order. Texture doesn't work particularly well because, when texture patterns are applied to the surfaces of objects in graphs (typically by using crosshatching, etc.), they tend to create a visually cluttered appearance. An added mark could only be used to represent a binary variable (i.e., one with only two potential values, such as female and male), for the added mark is either there or it isn't. It is also possible to use added marks that always appear in graphs by attaching them to the primary object that's being used to encode values, such as by always displaying a border around bubbles in a bubble plot, and by applying one of the other attributes in the list above (e.g., color intensity) to the border to encode a quantitative variable.

So, where does this leave us? Even though all 21 of these preattentive attributes can potentially be used to encode variables in graphs, only a few work well. As it turns out, however, this is not the only reason why a single graph can only effectively display a limited number of variables. The number of variables that we encode in a single graph is also affected by the fact that 1) certain visual attributes cannot be combined effectively in a single graph, 2) working memory can only handle three or at most four variables at a time, and 3) too many visual attributes tend to produce visual clutter. We'll consider those limitations next.

Effective Combinations of Visual Attributes

Some visual attributes can be combined in a single graph and some cannot. For example, 2-D horizontal position, 2-D vertical position, hue, and simple shape can work fairly well together in a scatter plot. On the other hand, as I've already pointed out, we cannot effectively combine hue and color intensity together in a single graph.

Another ineffective combination is the use of both line length and line width for separate variables. This is because length and width function as *integral* attributes. This means that, when they are combined, we perceive the result as area rather than as independent attributes of length and width. Imagine that we used the lengths of bars to encode one variable and the widths of bars to encode a second. We would preattentively perceive this combination as differences in the overall areas of bars, no longer independently as differences in the bars' lengths and widths. Although we could not perceive length and width as separate variables preattentively, we could do so with conscious effort, but it would be much slower.

Here's a list of the attributes that cannot be effectively combined:

- Line length and line width, because these attributes are integral
- Any attributes of color (e.g., hue and color intensity, hue and transparency, or color intensity and transparency)
- Size and color (either hue, intensity, or transparency), when the sizes of objects become tiny
- Shape and size, for we cannot effectively compare the sizes of objects that vary in shape (e.g., circles, squares, triangles, and stars)
- Shape and curvature, because curvature is an aspect of shape and changing the curve would change the shape
- Shape and line orientation, because only a few shapes, such as lines and rectangles, would make it easy to perceive and compare slopes

These attributes can certainly be combined in a single graph, but they cannot be combined effectively.

Limits of Working Memory

In the moment when we're thinking about things (i.e., while we're attending to them), information is held in working memory. This is different from long-term memory, which functions as a form of permanent storage for later retrieval. When you retrieve information from long-term memory, you pull it into working memory to think about and manipulate it in the moment. Working memory is volatile in that, once information is released from working memory to free up space for new information, it is forgotten unless we take time to rehearse it enough to store it in long-term memory. In addition to being volatile, working memory is extremely limited. As I've already mentioned, we can only hold from three to four chunks of information in working memory at a

time. Consider the number 417. Although it is composed of three digits, it can be held in working memory as a single chunk of information. While thinking about and comparing quantities, we could simultaneously hold the numbers 417, 25, and 5,003 in working memory as three discrete chunks. Data visualization is powerful, in part, because it allows us to chunk multiple values together in a way that expands the amount of information that can be simultaneously held in working memory. For example, the pattern formed by a line in a line graph that represents 12 monthly sales values can potentially be held in working memory as a single chunk (i.e., as the visual pattern formed by the line), whereas only three or four of those values could be held simultaneously in working memory when represented as numbers.

This limitation in the capacity of working memory plays a significant role in data visualization. When we view a graph for the purpose of reading and comparing values, the fact that we can only simultaneously hold up to three or four chunks of information in working memory limits the comparisons that we can make in any one moment. Fortunately, because a great deal of information is potentially there in front of our eyes, we can quickly swap information in and out of working memory as needed, but never hold more than four chunks at a time.

Here's the clincher. When multiple variables are represented by different visual attributes, we cannot chunk them together in working memory. For example, if we're viewing a bubble plot that uses 2-D horizontal position, 2-D vertical position, bubble size, and bubble color intensity to encode four variables in each bubble, each of those values is held in working memory as a separate chunk. If we applied additional visual attributes to those bubbles to encode more variables, we would still only be able to hold up to four at a time in working memory. Now, what if we want to compare one of those bubbles to another? If each bubble represents four values, totaling eight for two bubbles, we could only hold two values at a time for each bubble in working memory when making comparisons. This means that we would be forced to swap values in and out of working memory to compare more than two values per bubble. Consequently, even though we could encode more variables in a single graph using different visual attributes, it wouldn't expand our ability to consider them simultaneously. Even four variables per object exceeds the number that we could consider in any one moment when we're comparing objects to one another. As far as I know, no research studies have ever measured the efficiency gains or losses for various tasks (e.g., decoding the various values that are associated with an object, comparing objects of various types, etc.) that are associated with the number of variables that are encoded in a single graph. Given proper study, we might find ways to improve efficiency, but for now we must keep these limits in mind.

This limitation in the capacity of working memory, combined with the fact that most visual attributes do a relatively poor job of representing values in graphs, forces us to admit that any gains in efficiency that we're hoping to achieve by including more than a few variables in a single graph are wasted. It's worse than that, actually, for each additional visual attribute that we include in a graph potentially contributes to the appearance of clutter, which is our next topic.

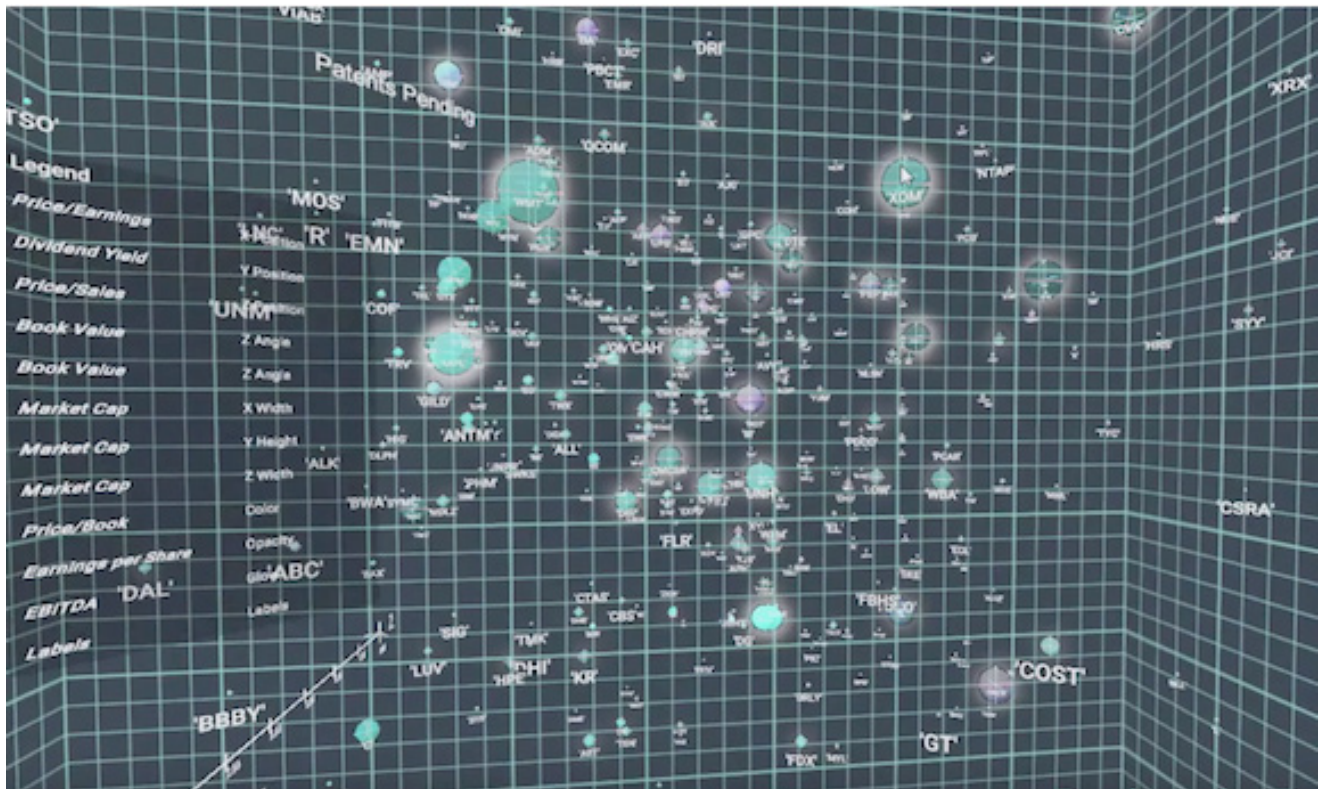
The Distraction of Visual Clutter

By clutter, I'm referring to the characteristics of a graph's appearance that are potentially messy looking and distracting when we're trying to focus on the particular attributes that we care about in the moment. For example, there is no doubt that having objects blink on and off at various speeds to encode a quantitative variable would make it almost impossible to attend to anything else. Even overly bright colors result in a cluttered appearance that is distracting. Every additional variable encoded by introducing another visual attribute to a graph comes with a perceptual cost. The cleaner and simpler the display, the easier it is to use.

When a chef chooses among the ingredients in her kitchen to cook a soup, her goal is not to combine as many ingredients as possible but instead to combine only those that are needed and to prepare them in the best way possible to create a pleasing culinary experience. Similarly, when we choose among the variables in a data set and display them in a particular way in a graph, our goal is not to squeeze as many variables as possible into it but to answer the question at hand in the most enlightening way. When visualizing data, we don't typically start with a single graph and then ask many questions about it. Instead, we start with questions, one at a time, and create graphs as needed to answer each in the best possible way.

Be very wary of data visualization vendors that promote their supposed ability to display a large number of variables in a single graph. More isn't better. Better is better. Only vendors that have taken the time to study visual perception and cognition can build data visualization tools that actually work. Unfortunately, relatively few vendors have done this, which is painfully obvious from the dysfunctional tools that most of them sell.

Data visualization vendors, especially newcomers, occasionally make the erroneous claim that their software can effectively visualize a large number of variables at once using separate visual attributes for each. I encountered the latest example of this recently when I read a press release about a new product named Immersion Analytics by the company Virtual Cove. These folks claim that, using their patents-pending techniques, they can effectively visualize up to 16 variables simultaneously. The following example includes 12 variables:



One of the arguments that Virtual Cove makes to promote their software is that to visualize a data set consisting of 16 variables using graphs that display only 4 variables each would require 1,820 graphs in total, which their software could replace with a single graph. They made this specific claim in an email to me, and they feature similar claims in their marketing efforts. It's probably quite persuasive to many people, for it has the air of mathematical certainty. As it turns out, however, it is neither accurate nor relevant. I'm not sure how they did the math, but it appears to be based on the invalid assumption that every possible combination of four-variables would need to be examined to compare each of the 16 variables to each of the others. That isn't the case. To see each of the 16 variables in relation to each of the other 15 using four-variable bubble plots, for example, would only require 35 graphs, not 1,820. Their figure is off by a factor of 52. The actual number of graphs that would be needed is less than 2% of the figure that they claim. Even if we were looking for correlations among 16 quantitative variables using scatter plots with only two variables each, that would only require a total of 120 graphs. In fact, a scatter plot matrix could be used to display all of these scatter plots at once. Even though this might require some scrolling around on the screen to examine every scatter plot, that wouldn't matter because we would only need to view one scatter plot at a time. A scatter plot matrix would provide insights that could never be achieved using a single graph that attempts to encode 16 variables using Virtual Cove's approach.

Given their egregious error, do you suspect that Virtual Cove might be making numbers up when they claim, as they do on their website, that their software can "increase productivity by up to 400x"? A four-hundred-fold

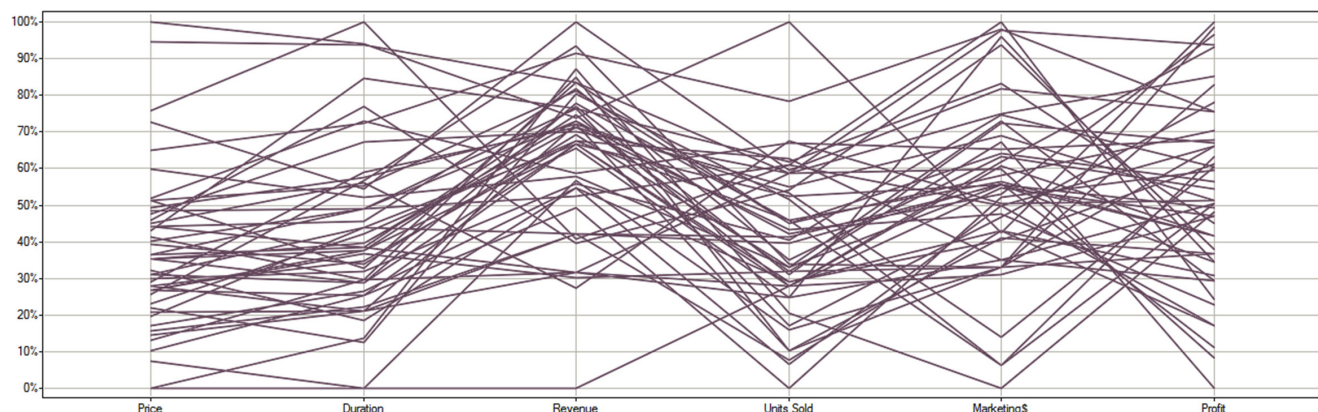
increase? Really? That means that if the conventional approach took one hour of time, their approach would reduce the work to nine seconds. Can you guess what their response was when I asked for evidence of this claim? You're right if you guessed that they didn't respond.

Encode Every Variable Using the Same Visual Attribute

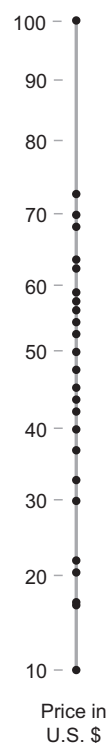
Instead of encoding each variable in a graph using a different preattentive attribute of visual perception, multiple variables can be displayed in a graph using the same attribute. Two types of graphs in particular were invented to use this approach for specific purposes: *parallel coordinates* and *table lenses*.

Parallel Coordinates

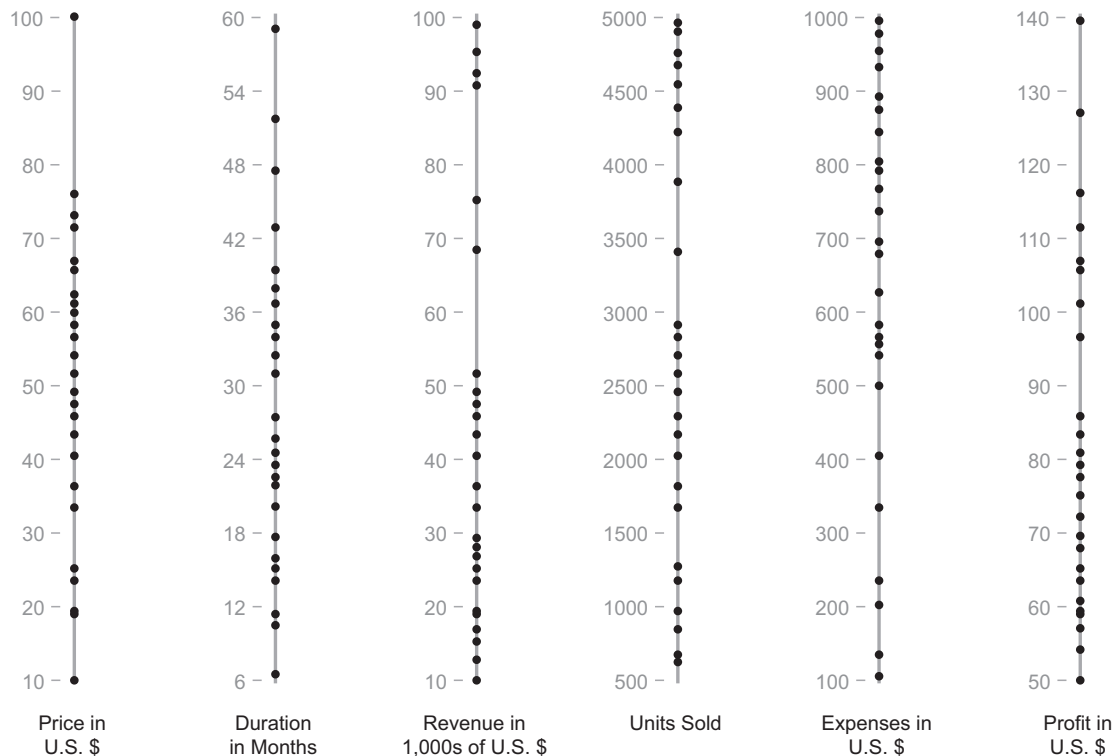
A parallel coordinates plot uses 2-D position, most often vertically along Y axes, to encode a series of variables. The example below displays six quantitative variables, each along its own Y axis.



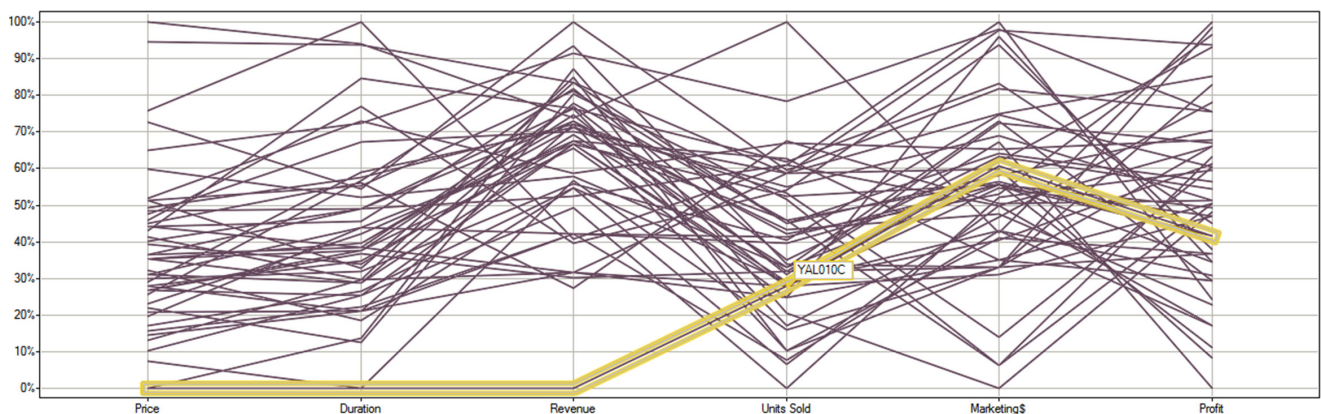
In case you're not familiar with parallel coordinates plots, let me briefly explain how they work. Let's begin by considering a single variable. In the example below, the prices in dollars for 25 products have been represented by positioning 25 dots along the Y axis. When each value is represented by a dot along a single quantitative scale in this manner to show how the values are distributed, the graph is called a *strip plot*.



Although strip plots are more typically arranged along the X axis, the Y axis can work just as well. When strip plots are arranged vertically, multiple strip plots can be placed side by side to display an entire series of variables, such as the six variables that appear below for the same set of 25 products.



So far, however, we cannot determine which dot represents which product. That would be useful if we want to determine how the products compare to one another across the entire set of six variables. To make this possible, a parallel coordinates plot would connect the dots for each product across each of the Y axes using a line. In the example below, which displays multivariate data for 50 products, a particular line is highlighted to feature a single product's multivariate profile.



(Note: In this example, rather than assigning a separate quantitative scale to each variable, the scales have been normalized by expressing each as percentages: the item with the lowest value is at the bottom with 0% and the one with the highest value at the top with 100%. Because the purpose of a parallel coordinates plot is not to decode individual values but instead to examine and compare multivariate patterns, the scales can be normalized in this manner without a loss of relevant information.)

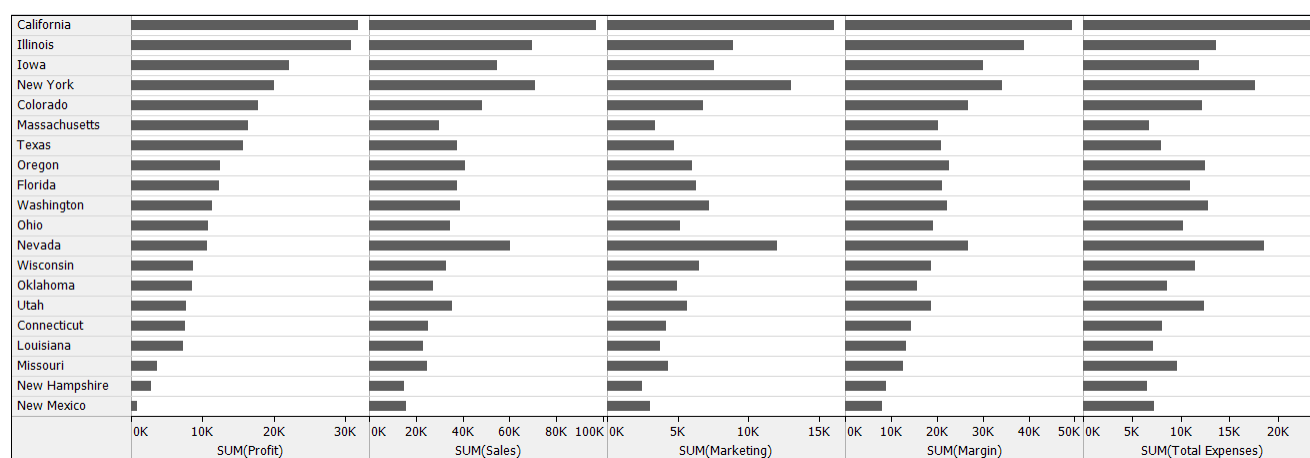
In this example, we have a single graph that displays six variables for 50 products, but a parallel coordinates

plot can include more variables and more than 50 items. As you might imagine, parallel coordinates plots can become complex and cluttered when they include many variables and items, but they can still be used to effectively compare complex multivariate profiles when properly designed, especially through the use of filtering and highlighting. Unlike graphs that encode variables using a different visual attribute for each, by encoding each variable in the same way (i.e., as 2-D vertical position), a parallel coordinates plot displays multiple variables in a way that our brains can read and interpret quite effectively. Because each one of an item's values is connected by a line, the pattern formed by that line can be held in working memory as a single chunk of information. When we hold that line in working memory, we are not holding each variable's value in memory, but that isn't necessary when we're trying to compare multivariate profiles, which we can do by simply comparing the patterns of multiple lines. For such a task, this approach to multivariate display is brilliant.

Without more thorough instruction in parallel coordinates, an example like the one above might appear overwhelming, so you might doubt the ability of these graphs to present complex multivariate data in a way that works for our brains. They do require extensive study and practice, which is one of the reasons why they are not more familiar, but they can definitely be worth the effort if you need to compare complex multivariate profiles. For a bit more explanation, I suggest that you read the newsletter article titled "[Multivariate Analysis Using Parallel Coordinates](#)" that I wrote back in 2006.

Table Lenses

A table lens display also uses a series of axes, one per variable, arranged side by side, but the arrangement is slightly different from parallel coordinates plots. Here's a simple example of a five-variable table lens display:

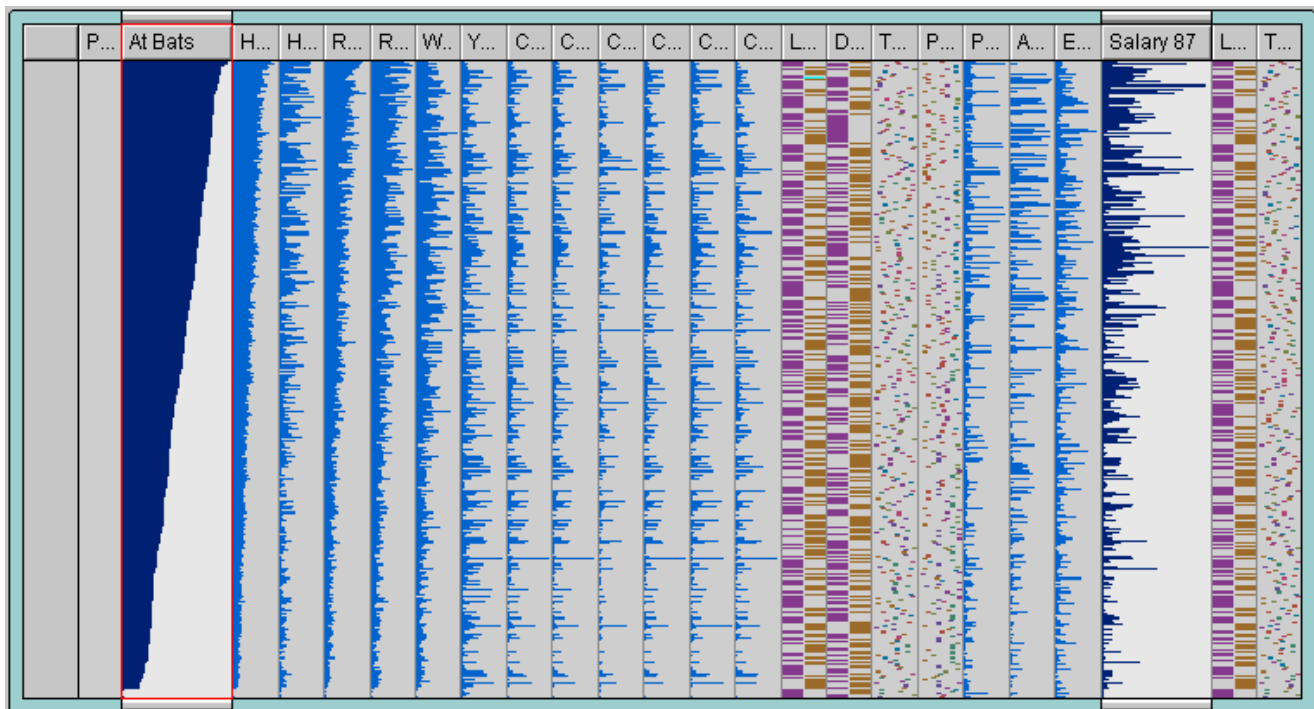


In this case, the Y axis host a categorical scale that labels the item for which quantitative data is being displayed, in this case U.S. states, and the X axes host independent quantitative scales, one per variable. When values are represented as bars, the horizontal position of each bar's end and the length of each bar both represent the same quantitative value. Unlike parallel coordinates, which are used to compare multivariate profiles, table lenses are used to look for potential correlations among several quantitative variables at once.

Notice in the example above that the states have been ranked from the highest value at the top to the lowest value at the bottom based on profit, the leftmost variable. Given this arrangement, we can now look at the arrangements of bars from top to bottom in each of the other columns to see if any of the other variables exhibit patterns that are similar to profit or are perhaps its inverse. If the arrangement of bars for one of the other variables roughly displays a pattern ranging from high values at the top to low values at the bottom, this tells us that it correlates with profit in a positive way. That is, as profit values per state decrease, values of sales also tend to decrease. If, on the other hand, sales roughly exhibit a pattern of low values at the top to high values at the bottom, this would tell us that it is still correlated with profit, but in a negative manner. That is, as profit values decrease, sales values tend to increase.

A table lens can provide a useful way to look for correlations among many variables at once. The example

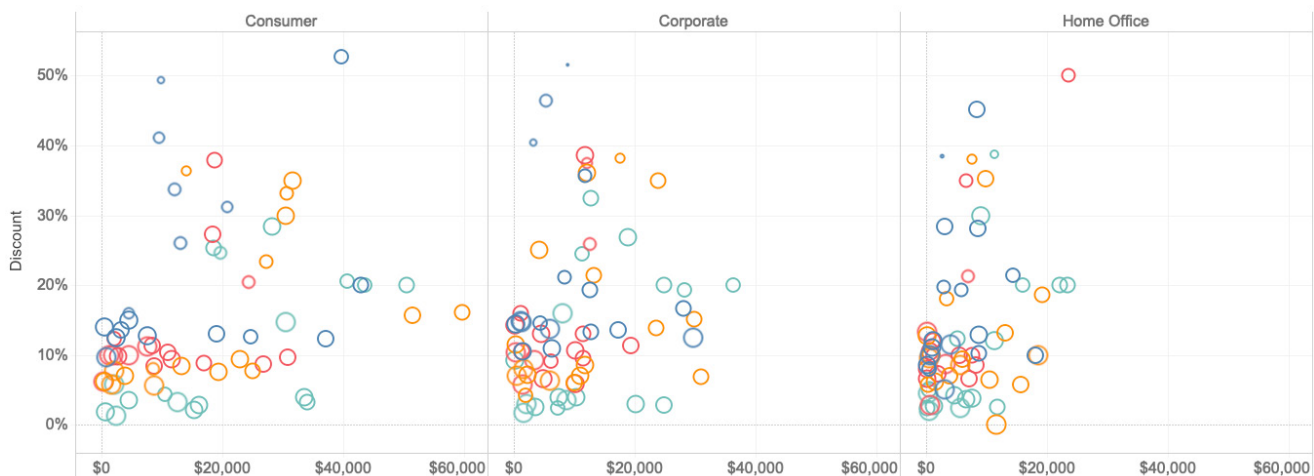
below, which was produced using a product that was actually called Table Lens from a company named Inxight, which unfortunately no longer exists, displays 23 variables worth of baseball statistics.



A table lens can display many variables in a single graph in a manner that works for our brains because it encodes each using the same visual attribute—one that we can perceive with ease.

Increase the Number of Variables Using Small Multiples

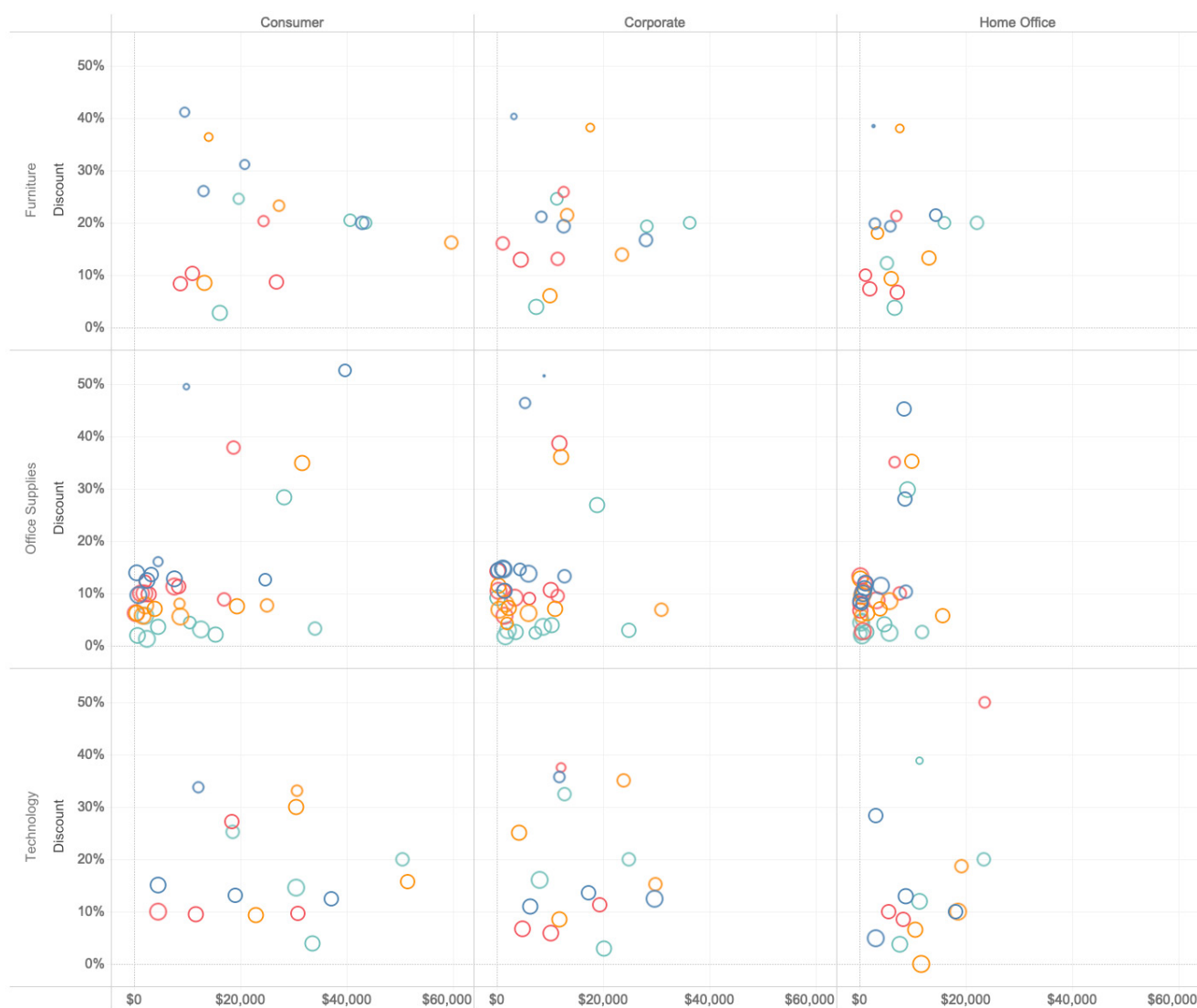
This final data visualization approach is used to increase the number of variables that can be simultaneously displayed. This approach goes by various names, but the most familiar is Edward Tufte's term *small multiples*. Back in the 1970s, both Edward Tufte and William Cleveland promoted the use of displays that combine several small graphs. Each graph works the same, but each displays data associated with a different categorical item. In the example below, three small graphs have been arranged side by side, and each displays data associated with a different customer segment: Consumer, Corporate, and Home Office. Other than this, the three graphs work exactly the same. Each displays sales revenue in U.S. dollars along the X axis, discount percentage along the Y axis, profit margin by bubble size, and geographical region by bubble hue.



The individual graphs in this example are already complex enough with four variables each, so it wouldn't work to display additional variables in them. The additional variable of customer segment, however, has been added to the display without overcomplication by presenting each customer segment in its own graph.

Many more graphs than the three that appear in the example above can be included in a small multiples display and they can be arranged on the screen in various ways. The example above arranges the small multiples horizontally in a single row, side by side, but they could also be arranged vertically, in a single column. A large series could also be wrapped across multiple columns and rows—an arrangement that William Cleveland called a *trellis display*.

Alternatively, a series of small multiples can be used to add two more categorical variables rather than just one. In the example below, each column of graphs still displays customer segments, but now each row displays product categories.

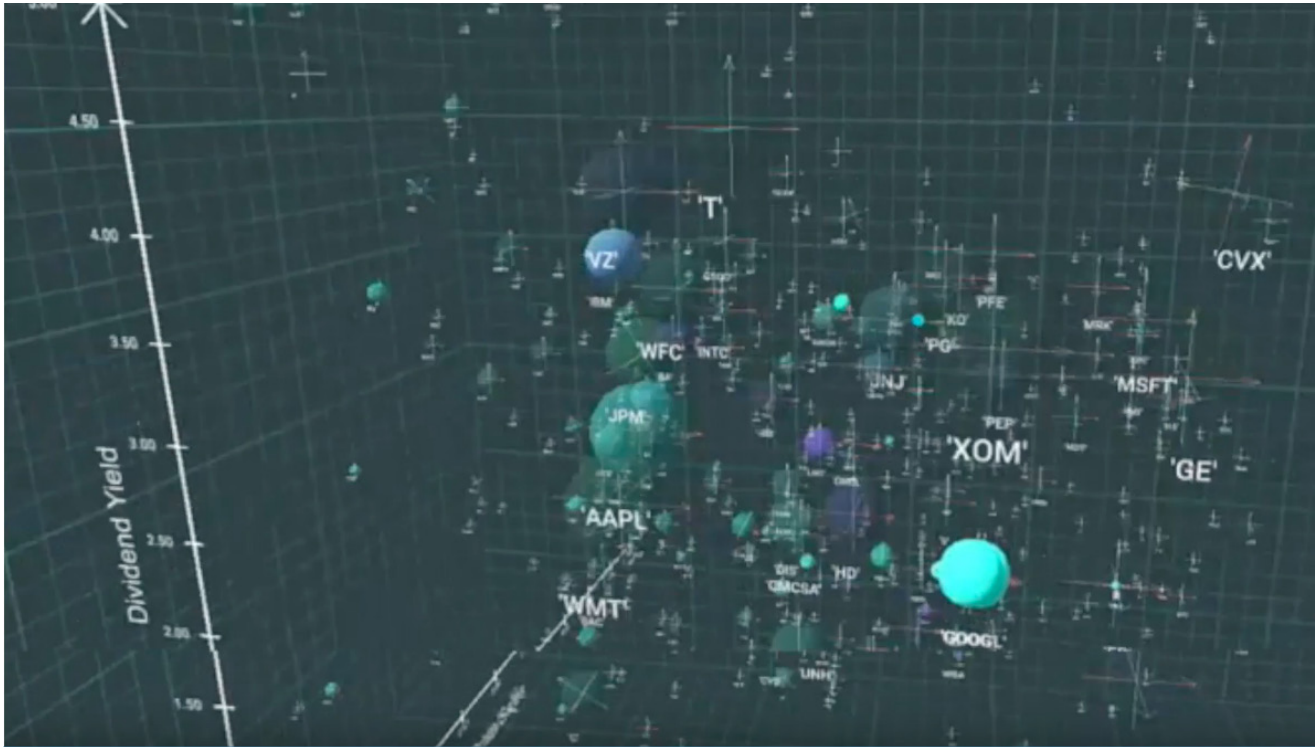


When small multiples are arranged in this way, with one variable along the rows and another along the columns, I call it a *visual crosstab*.

Even though a small multiples display consists of multiple graphs, because all the graphs work the same and are all visible at once, we can easily and quickly compare them to one another. If we know how to read one graph, we know how to read them all. This is a powerful way to increase the number of variables that can be simultaneously displayed beyond the number that you could include in a single graph that encodes variables using different visual attributes.

Conclusion

Wanting to break through our limitations is natural. We want to be better; we want to do more. We don't accomplish this, however, by ignoring our limitations. Ignorance is the path to delusion and dysfunction. Software vendors don't get any points for building and selling tools that simultaneously visualize a dozen or more variables in ways that don't work. When our limitations get in the way, we overcome them by using our brains to find real solutions. We always begin by understanding our limitations. Parallel coordinates plots, table lens displays, and small multiples are all innovations that demonstrate the merits of this approach. On the other hand, the graph below shows what happens when we simply ignore our limitations.



This graph only displays eight variables, half the number that the vendor, Virtual Cove, claims to support, and it's already a virtual cave of worthless effects. We can only see that a few of the spheres (i.e., 3-D bubbles) are much bigger than the rest and that one is much brighter as well. Imagine how much worse it would be if this graph attempted to display 16 variables rather than 8.

The potential for understanding that resides in our data should not be wasted by chasing pipe dreams. The path forward begins by understanding our limitations, not by pretending that they don't exist.

About the Author

Stephen Few has worked for over 35 years as an IT innovator, consultant, and teacher. In 2003, he founded the consultancy Perceptual Edge, which focused on data visualization for analyzing and communicating quantitative information. Today, he continues to dabble in data sensemaking and data visualization as diversions from a life of leisure. He is the author of six books:

- Show Me the Numbers: Designing Tables and Graphs to Enlighten, Second Edition
- Information Dashboard Design: Displaying Data for at-a-Glance Monitoring, Second Edition
- Now You See It: Simple Visualization Techniques for Quantitative Analysis
- Signal: Understanding What Matters in a World of Noise
- Big Data, Big Dupe: A Little Book about a Big Bunch of Nonsense
- The Data Loom: Weaving Understanding by Thinking Critically and Scientifically with Data

You can learn more about Stephen's work and access an entire library of articles at www.PerceptualEdge.com. Between articles, you can read Stephen's thoughts on the data visualization and data sensemaking in his two blogs: www.PerceptualEdge.com/blog and www.Stephen-Few.com/blog.