SAMPLE CHAPTER

Visualizing GRAPH DATA

Corey L. Lanum

MANNING

Visualizing Graph Data

by Corey L. Lanum

Chapter 1

brief contents

1 Getting to know graph visualization

This chapter covers

- Getting to know graphs as data models
- Why graphs are a useful way to think about data
- When to visualize graphs, and the node-link drawing concept
- Other visualizations of graph data and when they're useful

In December 2001, the Enron Corporation filed for what was at the time the largest ever corporate bankruptcy. Its stock had fallen from a high of $90 per share the previous year to $0.61, decimating its employees' pensions and shareholders' investments in it. The FBI's investigation into this collapse became the largest white-collar criminal investigation in history as they seized over 3,000 boxes of documents and 4 terabytes of data. Among the information seized were about 600,000 emails between key executives at the organization. Although the FBI took pains to read every email individually, the investigators recognized that they were unlikely to find a smoking gun—people committing complex financial fraud seldom disclose their actions in written form. And in 2001, emails were only starting to become

the primary means of internal communications; lots of information was still exchanged via phone calls.

In addition to looking at the text of individual emails, the FBI also wanted to uncover patterns in the communications, perhaps in an attempt to better understand who the decision makers were within Enron or who had access to a lot of the information internal to the company. To do this, they modeled the Enron emails as a *graph*.

A *graph* is a model of data that consists of *nodes*, which are discrete data elements (such as people), and *edges*, which are relationships between nodes. The graph model brings to the forefront relationships that may be hidden in tabular views of the same data and illustrates what is most important. By making those relationships between the data elements a core part of the data structure, you can identify patterns in the data that wouldn't otherwise be apparent. But building graph data structures is only half the solution to pattern recognition. This book will teach you how to visualize graphs using interactive node-link visualization diagrams, and by the end, you'll be able to create your own dynamic, interactive visualizations using a variety of tools available today.

In this chapter, I'll go a little deeper into the concept of a graph and graph history and uses, and talk about various techniques used to visualize graph data. Subsequent chapters build on this framework by introducing concrete examples of graph visualizations and the data they're based on and discuss various techniques for creating useful visualizations.

1.1 Getting to know graphs

Graphs are everywhere. As long as you're interested in how items can be related to each other, there's a graph somewhere in your data. In this section, I'll walk you through what a graph is and what can be gained from visualizing graphs.

1.1.1 What is a graph?

As described previously, a graph—also called a *network*—is a set of interconnected data elements that's expressed as a series of nodes and edges.

In the common definition of a graph, edges have exactly two endpoints, no more. In some cases, those two endpoints can be the same node if a node links to itself. An edge (also known as a link) can take one of two forms:

- *Directed*—The relationship has a direction. Stella owns the car, but it doesn't make sense to say the car owns Stella.
- *Undirected*—The two items are linked without the concept of direction; the relationship inherently goes both ways. If Stella is linked to Roger because they committed a crime together, it means the same thing to say Stella was arrested with Roger as it does to say Roger was arrested with Stella.

In figure 1.1, you see an example of a directed link with properties.

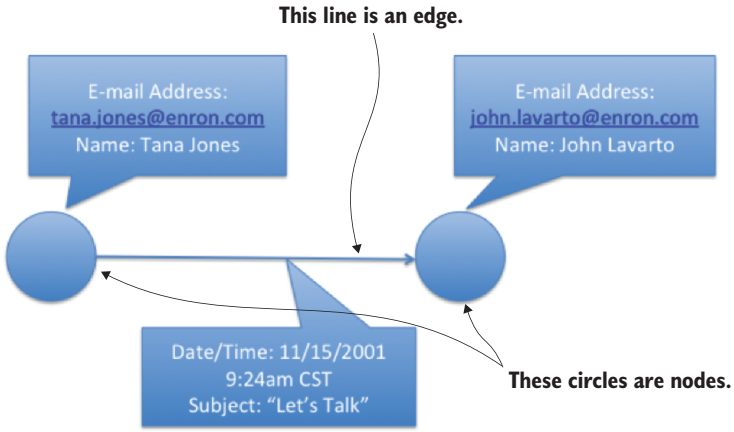

Figure 1.1 A property graph of a single email between Enron executives. The two nodes are the sender and recipient of the email, and the directed edge is the email.

Both nodes and edges can have *properties*, which are key-value pairs—lists of properties and values, describing either the data element itself or the relationship. Figure 1.2 is a simple property graph showing that Stella bought a 2008 Volkswagen Jetta in September 2007 and sold it in October 2013. Modeling it as a graph highlights that Stella had a relationship with this car, albeit temporarily.

An email is a relationship, too, between the sender and the recipient. The properties of the nodes are things like email address, name, and title, and the properties of the relationship are the date/time it was sent, its subject line, and the text of the email.

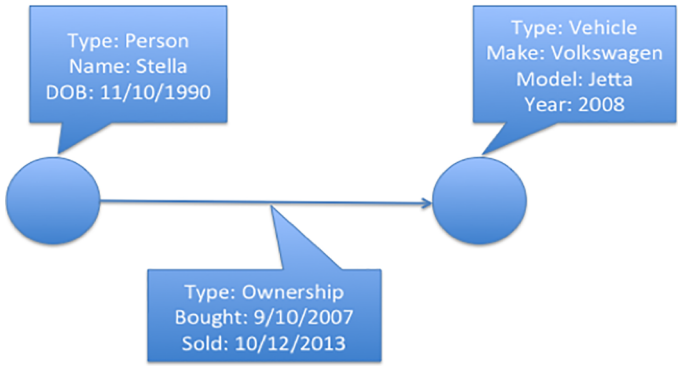

Figure 1.2 A simple property graph with two nodes and an edge. Stella (the first node) bought a 2008 Volkswagen Jetta (the second node) in September 2007 and sold it in October 2013. Modeling it as a graph highlights that Stella had a relationship with this car (the edge).

To prove conspiracy, the FBI was interested in all the emails sent among the Enron executives, not just a single one, so let's add some more nodes to represent a larger number of emails sent during a specified period of time, as shown in figure 1.3.

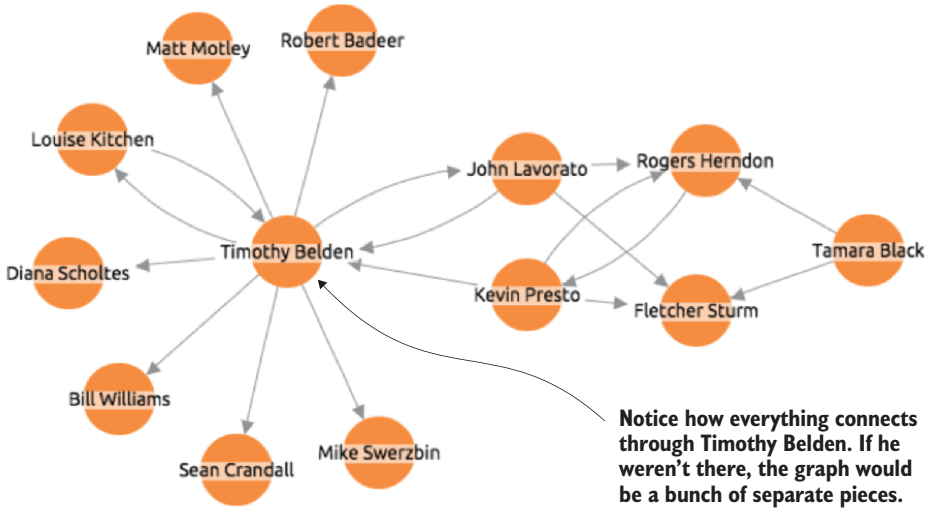

Figure 1.3 A graph of some of the Enron executives' email communications. You can easily see that Timothy Belden is a hub of communication in this segment of Enron, sending and receiving email from many other executives.

Figure 1.3 is a *directed graph* because it matters whether Kevin Presto sent an email to Timothy Belden or received one—there's a big difference between sending and receiving information when you're investigating who knew what when. The arrowheads on the edges show that directionality: Kevin Presto sent an email to Timothy Belden, but Timothy Belden didn't reply, indicating they may not have been close associates or they may have spoken offline. As we start to add more data to the graph, you can see the value of graphs—patterns become apparent. In this example, we can easily see that Timothy Belden is a hub of communication in this segment of Enron, sending and receiving email from many other executives.

1.1.2 A bit of theory

Graph theory began early in the eighteenth century with the Seven Bridges of Königsberg problem. In Königsberg, Prussia (now Kaliningrad, Russia), it was a common parlor game to try to determine a route that would allow someone to pass over all seven bridges over the Pregel River exactly once without passing over any bridge twice. (Go ahead and give it a shot using the map of the city, shown in figure 1.4, and see if you can prove three centuries of mathematicians wrong.)

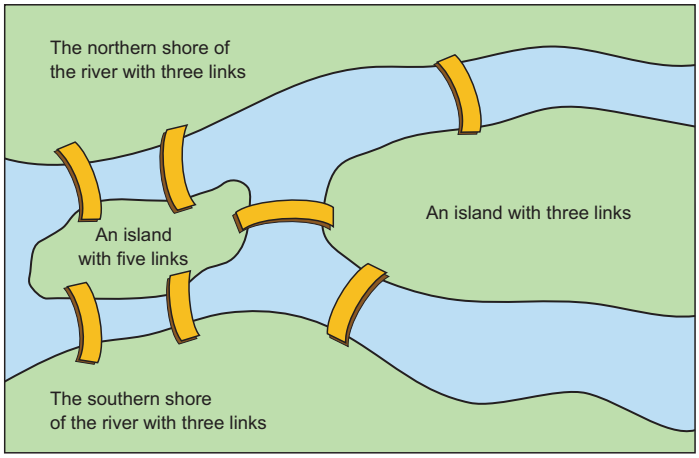

Figure 1.4 The Seven Bridges of Königsberg problem. Using this map of the bridges of Königsberg, Prussia, try to draw a route that reaches each area of the city but never crosses the same bridge twice.

Leonhard Euler proved this problem unsolvable by abstracting the regions of the city into individual points and the bridges as paths between those points, as you can see in figure 1.5.

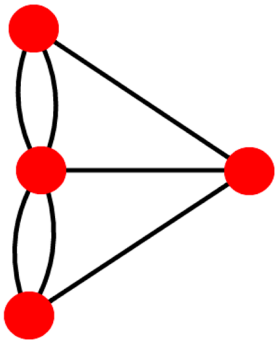

Figure 1.5 Seven bridges and four land areas of Königsberg as a graph. In this graph, nodes denote the land masses bordering the Pregel River and the two islands in its middle. Edges represent the bridges connecting the two islands and two shorelines.

Each land area of Königsberg is indicated by a point, and the bridges are the lines that connect the points. This is a graph, just like the Enron graph. The graph model in figure 1.5 makes it easier to see that nodes with an even number of links can be navigated easily (you can enter and exit using two different links), but nodes with an odd number of links can only be either the beginning or the end of a path (this is obvious for a node with only one link, but you can also see it applies to three, five, and so on). The number of links from a node is called that node's *degree*. The Königsberg bridge problem can now be proven solvable only if at most two nodes have odd degrees and the rest have even degrees. This diagram doesn't satisfy that condition, and therefore it's impossible to cross each bridge only once; so graph theory has answered a problem

previously considered intractable. The principle of nodes with even and odd degrees applies to all graphs, not just the Königsberg bridge problem.

1.1.3 Introducing the graph data model

Graphs are interesting mathematical constructs, and many academic mathematicians have spent their entire careers studying the domain. But the purpose of this book is to describe how graphs can be derived from data and how they can be presented to non-mathematicians so that they can better understand the data. Let's go back to the Enron example. I chose to model the data such that the nodes were the employees at Enron and the emails were represented as links between them, but that's not the only graph model that could be derived from this data. That model, and the resulting visualization, shows who is communicating with whom but ignores basic data about the email itself. And it fails to take into account possible interesting information such as forwarding of emails or sending a single email to multiple people, some of whom may be CC'd or BCC'd.

> **DEFINITION** A *visualization* is any method of using imagery to convey a point. With relation to computer graphics, it typically means finding a way to show large amounts of data in a single view. Creating images to show graph data is called *graph visualization*.

In this case, you may want to consider the email itself a node. Figure 1.6 shows the sender and the recipients of an email with the subject line "Let's Talk," sent among Enron executives. The nodes are executives from Enron, and the edges represent how they received the email (whether they were in the To:, CC:, or BCC: fields).

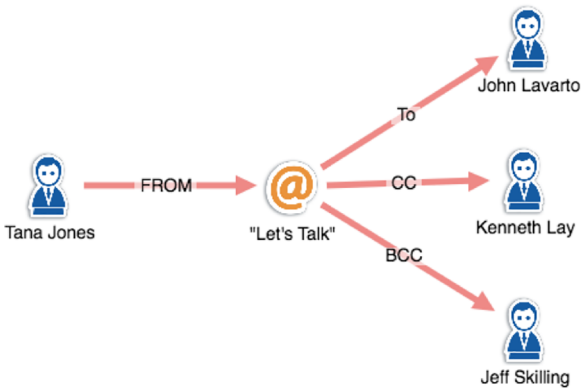

Figure 1.6 Graphing a single email at Enron

Consider the following very simple table. Table 1.1 consists of only two columns: a list of names and the countries where those names are used. In the United States, the Social Security Administration releases a similar table each year showing the

popularity of first names given to babies, as measured by new applications for Social Security numbers.

Table 1.1 A list of names and countries where those names are common

Name	Country
Joe	United States
Juan	Mexico
João	Brazil
Jean	France
Antoine	France
Jean	United States
Ignacio	Mexico
João	Portugal

This can be modeled as a graph—show each name as a node and each country as a node. A link between name and country nodes exists if the name and country appear in the same row. The result looks like figure 1.7.

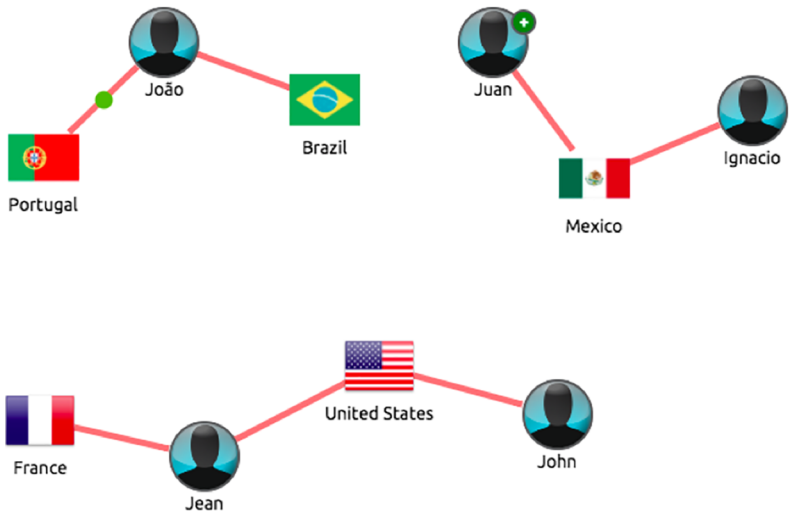

Figure 1.7 Using a graph to illustrate a table of name/country pairs

This graph tells us more than is immediately obvious from the table, namely, that *Jean* is used both in France and the United States. It also shows that *João* is used in Brazil and Portugal, but no other names are associated with those countries. As you can see, you can generate graph models from even the simplest data set, but typically you're

going to want properties on the nodes, the links, or both. In this case, you may want the frequency of a name's use in a country as a property of the link and perhaps whether that name is typically used by males or females.

1.1.4 When are graphs helpful?

Now that you understand what graphs are, why would you use them? Although there are certainly cases where a graph model wouldn't be appropriate—a long key-value pairing comes to mind—they can be very useful when there are relationships between your data elements. If the nodes are connected to one another somehow, and those connections are as important as the data itself, then a graph is a useful model for understanding the data. For example, in a list of financial data, it may be important to look at the data in aggregate, say in a budget where you're interested in the total amount spent in a set of categories. A graph would be counterproductive in this instance, because you aren't looking for connections within this data. You care only about the bottom line. But in the same set of data, if you're interested in the transactions embedded in the data—for example, which consumers are spending money at which merchants, and which merchants are using which banks—then a graph would be a very useful model for storing and visualizing that data.

VALUE OF GRAPHS

Graphs can be incredibly useful, but there's a danger in overusing them. Many people, when first exposed to graph concepts, begin to see graphs everywhere, in every data set, but a graph can sometimes obscure the meaning of the data.

Graphs are a good choice in the following situations:

- *The links between items are not obvious.* For example, linking someone's first name and last name together is unlikely to be useful unless you're looking at the relationship of the names when they are independent of each other. "How many 'Coreys' drive black cars?"
- *There's a structure embedded in the data.* If every link has unique end points with no other links, then the graph is a bunch of disconnected links that doesn't answer any interesting question.
- *There are at least some properties on the nodes.* If a data set doesn't have any properties, then it may create a pretty picture when drawn, but it's impossible to tell what you're looking at.

Figure 1.8 shows a graph model that's not very helpful. It represents the data found in the back of a road atlas that shows the mileage and driving times between city pairs.

In reality, every city in North America is connected to every other city via roads, and it's unlikely that the atlas browser wants to bother adding up the various segments and city pairs necessary to drive from Richmond to Buffalo; they just want to know how far it is and how long it takes to get there. In this case, they might prefer an association matrix, which is a different way of representing graphs.

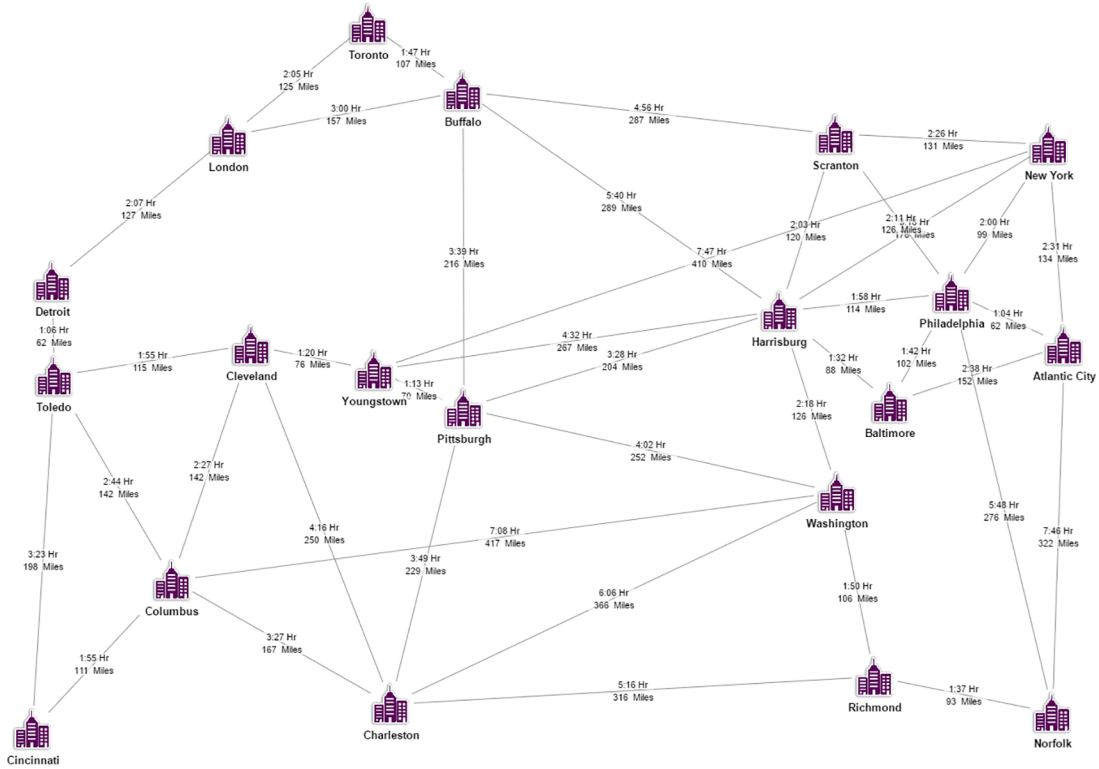

Figure 1.8 A graph of driving times between North American cities. A graph is not an effective way of presenting this type of data because in reality everything is connected to everything else.

ASSOCIATION MATRIX An association matrix is a table displaying the node names as both columns and rows. For each pair of nodes that has a link between them, the cell of intersection is either marked or filled in with a property value. In an undirected graph, the values will be repeated, because each node pairing appears twice: once with the first node as a row and the second as a column, and then vice versa.

The table shown in figure 1.9 is an association matrix presenting similar data on the distance between city pairs.

Sets of data where the relationships between the data elements are the most important feature are the most useful to model as a graph. Graphs work best when you have a key component of analysis. In this section, you've had a brief introduction to the graph data model, how tabular data can be represented as a graph (in section 1.1.3, "Introducing the graph data model"), and when graphs might be useful. In the next section, we'll discuss how and when to visualize graphs, that is, draw pictures of this data model on paper or on computer screens.

	Atlanta	Boston	Chicago	Dallas	Denver	Houston	Las Vegas	Los Angeles	Miami	New Orleans	New York	Phoenix	San Francisco	Seattle	Washington D.C.
Atlanta		1095	715	805	1437	844	1920	2230	675	499	884	1832	2537	2730	657
Boston	1095		983	1815	1991	1886	2500	3036	1539	1541	213	2664	3179	3043	440
Chicago	715	983		931	1050	1092	1500	2112	1390	947	840	1729	2212	2052	695
Dallas	805	1815	931		801	242	1150	1425	1332	504	1604	1027	1765	2122	1372
Denver	1437	1991	1050	801		1032	885	1174	2094	1305	1780	836	1266	1373	1635
Houston	844	1886	1092	242	1032		1525	1556	1237	365	1675	1158	1958	2348	1443
Las Vegas	1920	2500	1500	1150	885	1525		289	2640	1805	2486	294	573	1188	2568
Los Angeles	2230	3036	2112	1425	1174	1556	289		2757	1921	2825	398	403	1150	2680
Miami	675	1539	1390	1332	2094	1237	2640	2757		892	1328	2359	3097	3389	1101
New Orleans	499	1541	947	504	1305	365	1805	1921	892		1330	1523	2269	2626	1098
New York	884	213	840	1604	1780	1675	2486	2825	1328	1330		2442	3036	2900	229
Phoenix	1832	2664	1729	1027	836	1158	294	398	2359	1523	2442		800	1482	2278
San Francisco	2537	3179	2212	1765	1266	1958	573	403	3097	2269	3036	800		817	2864
Seattle	2730	3043	2052	2122	1373	2348	1188	1150	3389	2626	2900	1482	817		2755
Washington D.C.	657	440	695	1372	1635	1443	2568	2680	1101	1098	229	2278	2864	2755	

Figure 1.9 The back page of a road atlas. City names are represented as both columns and rows, and the cell of intersection indicates the distance between them.

1.2 Getting to know graph visualization

Why does visualizing graph data make it easier to understand? There are two reasons. Humans are intuitively visual creatures, and it's almost impossible to think of a model—any sort of model—and not picture how it looks. Ernest Rutherford developed the model of the atom in 1909 that we're all familiar with, a nucleus of protons and neutrons with electrons whizzing in orbits around the nucleus. This model was quickly replaced by Erwin Schrodinger's more accurate model based on quantum mechanics, but Rutherford's model is still, 90 years later, the one that the public embraces. Why? Because it can be pictured. The Schrodinger model is more accurate, but it's a mathematical concept, not a visual one, and therefore has never acquired broad public appeal. Data is the same way; unless we can show our audience what we're talking about, they won't recall it. Visualization helps bridge that gap and allows the decision-makers to understand the data.

I described the property graph model, with nodes, edges, and properties, in section 1.1, but the topic of this book is graph visualization. The sole reason for data collection is to make better-informed decisions based on the data, so it's important to provide a useful way of accessing it. And with graph data, that typically means drawing the graph.

Although there are many different methods of graph visualization, and I'll briefly discuss some of them, the focus of this book is on node-link visualization. This is not

to say that other visualizations are never useful, but node-link visualizations tend to have the broadest appeal regardless of the data source and require the least amount of technical knowledge to understand. I've been using node-link visualization so far in this chapter, and it's just what it sounds like. Nodes are points or polygons or icons, and links are lines connecting those points. Node-link diagrams are almost always drawn on a 2D plane and almost never three-dimensional. An important aspect of the node-link diagram is that a node's location doesn't tell you anything interesting about that node, although there are different ways of positioning nodes that can reveal some useful information based on location. Nodes are placed solely for convenience and readability, which makes this quite different from a Cartesian scatter plot, for example. An effect of this is that layout, or how nodes are arranged on the chart, becomes much more important. Two charts with identical data but different layouts can imply different things to the human eye.

1.2.1 When to visualize graphs

There are two reasons to visualize graphs, both of which are important:

- The first is to better understand the structure of the data that you have.
- The second purpose of visualization is to expose a broader audience to the data connections.

VISUALIZING GRAPH DATA STRUCTURE

The visualization in figure 1.10 shows the structure of the sales database and how its elements are connected to each other, but not the connections between individual employees and products.

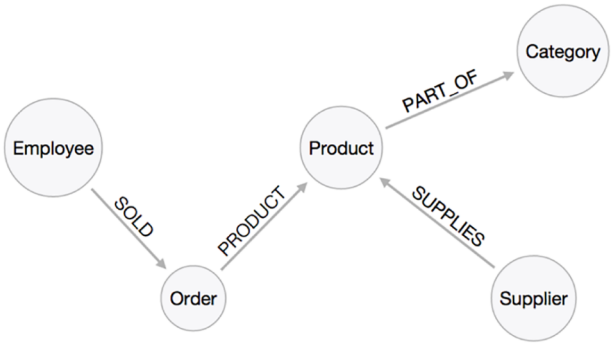

Figure 1.10 A sales database showing connections between different data types

With regard to the first purpose, understanding the structure: in a data set, what sorts of things are linked to other things? Many of the diagrams you'll see from graph databases are designed to illuminate this structure.

In this example, the structure of a sales database is visualized. Suppliers supply products that are members of categories. Employees take orders that consist of products. To a data scientist or an application engineer, this view is very important; it helps define the model of the data, how it's stored, and how users will interact with it. Get this wrong, and it can be a very time-consuming and expensive process to fix.

DRAWING YOUR GRAPH DATA

The second purpose is to visualize the data inside your data set. In this case, you're not interested in the categories of data but the actual relationships among the data elements themselves. Instead of saying "Employees sell orders that consist of products," you can get specific and say "Brad from Home Depot sold me a Husqvarna chainsaw." Then you can get more specific—who else is buying Husqvarna chainsaws from Home Depot? What other products are included in those orders? The visualization allows you to better understand the connections embedded within the data itself.

A key reason that graph visualization is important is it gives a visual interface to data discovery. Although much of the big data revolution of the past decade has been in understanding trends in the aggregate data, it's equally important to discover previously unknown connections and relationships between the individual data elements. A dashboard will be unlikely to show this, but a graph will allow the user to explore the data and discover these patterns visually. I'll discuss this more in chapter 6.

1.2.2 Alternative graph visualizations

The node-link visualization isn't the only way to display a graph, although it's the main focus of this book. In section 1.1, I mentioned the association matrix, where nodes are represented by columns and rows, and a mark (or a value) in the cell shows that there's a relationship between those two nodes. The road atlas in figure 1.9 is a good example. I find the association matrix very useful for creating or editing graph data, but not as helpful for explaining it to others. Next, I'll show you some examples of visualizations that are better than node-link visualizations for certain types of data.

CIRCLE PLOT

If the primary goal is to show aggregate links from groups of nodes, as opposed to individual data, something like the *circle plot* may make more sense. A good example is found at http://www.global-migration.info, which shows global migration between and within the six populated continents; see figure 1.11. Their data is a graph of migration patterns from each country to each other country, but a node-link diagram of all this data would be busy and wouldn't show aggregate patterns as well as the circle plot, so it was a good choice.

Figure 1.11 A graph from www.global-migration.info/. This stylish graph illustrates migration patterns among people from all six inhabited continents.

HIVE PLOT

Another example of an alternative to the node-link diagram is the *hive plot*, shown in figure 1.12. As mentioned earlier, node-link graph visualizations focus on individual data elements and the connections among them. While useful, they can fail to identify and communicate connections among different types or groups of data elements. The hive plot can be especially useful when trying to understand structure in very large networks, in tens or hundreds of thousands of nodes or links, because it differentiates nodes into three or more types and aligns them from the center of the chart on an

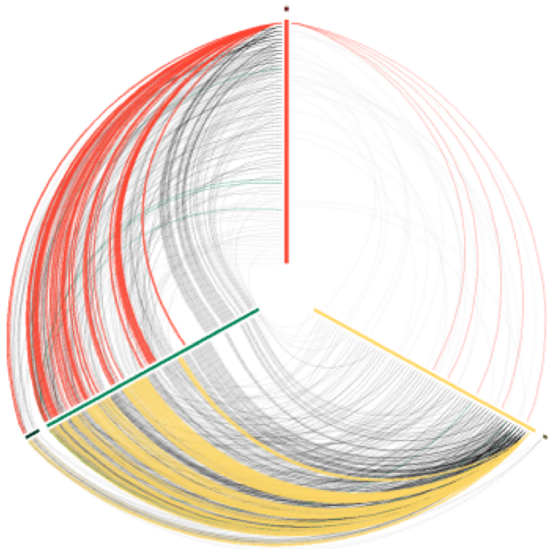

Figure 1.12 Hive plot of E. coli bacteria at http://www.hiveplot.net by Martin Krzywinski. Notice the large number of connections of the leftmost group but the fewer connections between the top and right.

axis. Links between elements of different types are drawn as curved lines around the center of the chart. This can allow a viewer to visually differentiate between two types that are tightly linked versus those that are weakly linked. It doesn't display links between elements in the same type and makes a drill-down, to look at subsets of the data, much more difficult.

There are many benefits to using node-link diagrams to better understand data, but at other times other visualizations are more helpful. In general, node-links are valuable when you want to focus on specifics, and are less helpful when you're interested in aggregates, as you just saw.

SANKEY DIAGRAM

Another useful visualization is the Sankey diagram, which is designed to plot flows of something (money, people, energy, and so on) from left to right in the chart. With traditional graph data, you're most interested in the relationship between the two nodes and any properties of that relationship, but a Sankey diagram is designed to highlight aggregate amounts between categories of nodes. The example in figure 1.13, from the International Energy Agency (https://www.iea.org/), shows the world's energy usage from raw materials on the left to consumed energy in the form of electricity or fuel on the right, with the intermediate steps in between. A node-link visualization would show individual gasoline refineries linked to individual oil fields as inputs and linked to individual gasoline wholesalers as outputs, but that's irrelevant detail if you're ultimately interested in the proportion of energy derived from various sources. The

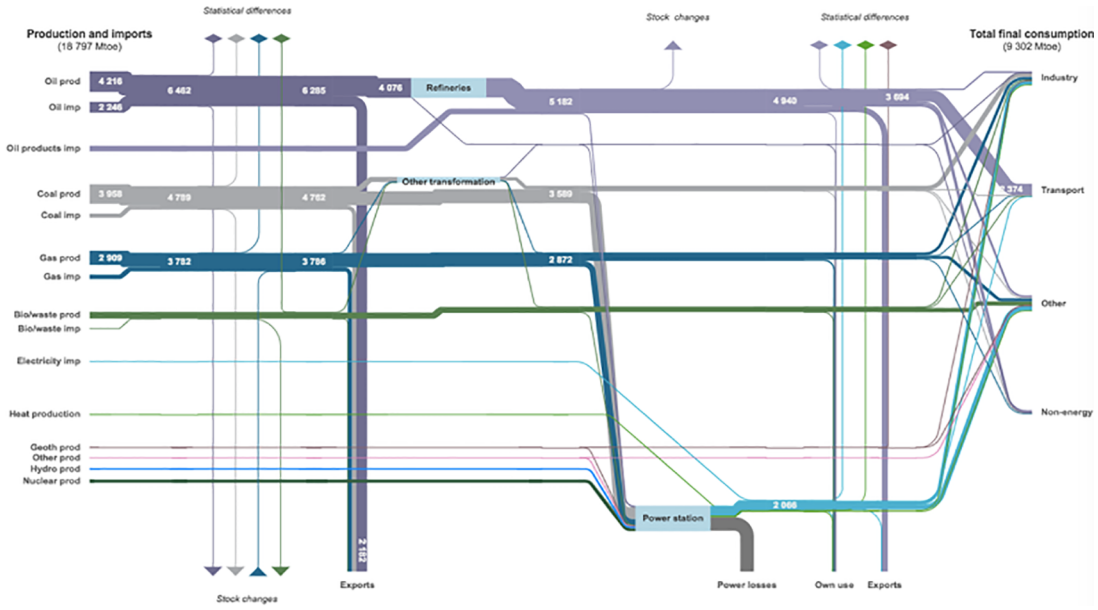

Figure 1.13 A Sankey diagram showing world energy consumption and the sources. See www.iea.org/Sankey for the larger view.

Sankey diagram makes clear what proportion of refined oil goes to transportation fuel versus electricity generation versus plastics, something that would be hard to see in a more traditional node-link visualization.

1.3 Summary

In this chapter, I introduced you to the definition of a graph and discussed the benefits of thinking about data in terms of nodes and links. I also emphasized when this is beneficial and when you'd see more benefit from a tabular view of data. I also touched on the history of graph visualization and the reasons why drawing a picture of your data can be enlightening. Although most of this book focuses on node-link visualizations, I mentioned a few other styles of visualization that can be helpful in certain circumstances.

- A graph is a model of data emphasizing the connections in the data.
- The graph model can be created from any data set where items share a common property. Some are more useful than others.
- Graph data can come from any source, not solely graph databases.
- The node-link diagram is the most common way of presenting and communicating graph data.
- Graph visualization can serve two purposes: it allows you to explore data and expose connections, and helps you communicate data connection findings to others.
- There are many other ways of displaying graph data that don't rely on the node-link diagram. Most of them help with looking at the structure of larger networks and not finer details.

PROGRAMMING/DATA SCIENCE

Visualizing GRAPH DATA

Corey L. Lanum

Assume you are doing a great job collecting data about your customers and products. Are you able to turn your rich data into important insight? Complex relationships in large data sets can be difficult to recognize. Visualizing these connections as graphs makes it possible to see the patterns, so you can find meaning in an otherwise overwhelming sea of facts.

Visualizing Graph Data teaches you how to understand graph data, build graph data structures, and create meaningful visualizations. This engaging book gently introduces graph data visualization through fascinating examples and compelling case studies. You'll discover simple, but effective, techniques to model your data, handle big data, and depict temporal and spatial data. By the end, you'll have a conceptual foundation as well as the practical skills to explore your own data with confidence.

What's Inside

- Techniques for creating effective visualizations
- Examples using the Gephi and KeyLines visualization packages
- Real-world case studies

No prior experience with graph data is required.

Corey Lanum has decades of experience building visualization and analysis applications for companies and government agencies around the globe.

To download their free eBook in PDF, ePub, and Kindle formats, owners of this book should visit
www.manning.com/books/visualizing-graph-data

"Shows you how to solve visualization problems and explore complex data sets. A pragmatic introduction."
—John D. Lewis, DDN

"Excellent! Hands-on! Shows you how to kick-start your graph data visualization."
—Rocio Chongtay
University of Southern Denmark

"A clear and concise guide to both graph theory and visualization."
—Jonathan Suever, PhD
Georgia Institute of Technology

"Great coverage, with real-life business use cases."
—Sumit Pal, Big Data consultant

ISBN-13: 978-1-61729-307-8
ISBN-10: 1-61729-307-5
54499
9 781617 293078

MANNING $44.99 / Can $51.99 [INCLUDING eBOOK]