1.1 What is data science?

Data science is the practice of using data to try to understand and solve real-world problems. This concept isn't exactly new; people have been analyzing sales figures and trends since the invention of the zero. In the past decade, however, we have gained access to exponentially more data than existed before. The advent of computers has assisted in the generation of all that data, but computing is also our only way to process the mounds of information. With computer code, a data scientist can transform or aggregate data, run statistical analyses, or train machine learning models. The output of this code may be a report or dashboard for human consumption, or it could be a machine learning model that will be deployed to run continuously.

If a retail company is having trouble deciding where to put a new store, for example, it may call in a data scientist to do an analysis. The data scientist could look at the historical data of locations where online orders are shipped to understand where customer demand is. They may also combine that customer location data with demographic and income information for those localities from census records. With these datasets, they could find the optimal place for the new store and create a Microsoft PowerPoint presentation to present their recommendation to the company's vice president of retail operations.

In another situation, that same retail company may want to increase online order sizes by recommending items to customers while they shop. A data scientist could load the historical web order data and create a machine learning model that, given a set of items currently in the cart, predicts the best item to recommend to the shopper. After creating that model, the data scientist would work with the company's engineering team so that every time a customer is shopping, the new machine learning model serves up the recommended items.

When many people start looking into data science, one challenge they face is being overwhelmed by the amount of things they need to learn, such as coding (but which language?), statistics (but which methods are most important in practice, and which are largely academic?), machine learning (but how is machine learning different from statistics or AI?), and the domain knowledge of whatever industry they want to work in (but what if you don't know where you want to work?). In addition, they need to learn business skills such as effectively communicating results to audiences ranging from other data scientists to the CEO. This anxiety can be exacerbated by job postings that ask for a PhD, multiple years of data science experience, and expertise in a laundry list of statistical and programming methods. How can you possibly learn all these skills? Which ones should you start with? What are the basics?

If you've looked into the different areas of data science, you may be familiar with Drew Conway's popular data science Venn diagram. In Conway's opinion (at the time of the diagram's creation), data science fell into the intersection of math and statistical

knowledge, expertise in a domain, and hacking skills (that is, coding). This image is often used as the cornerstone of defining what a data scientist is. From our perspective, the components of data science are slightly different from what he proposed (figure 1.1).

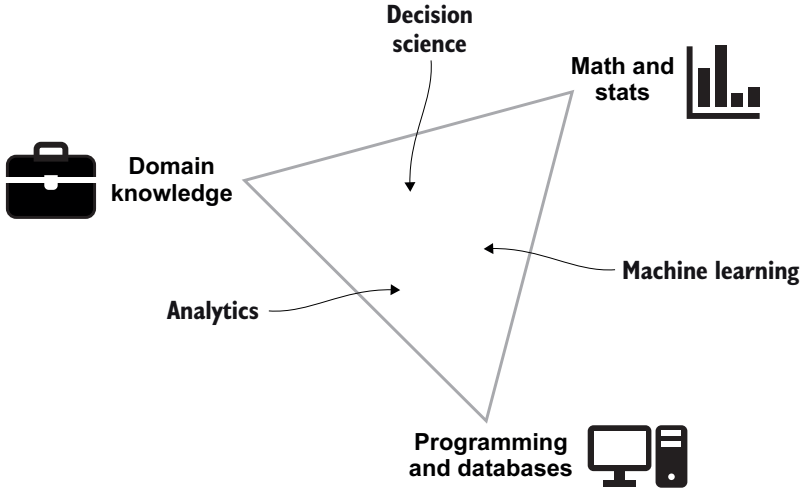

Figure 1.1 The skills that combine to make data science and how they combine to make different roles

We've changed Conway's original Venn diagram to a triangle because it's not that you either have a skill or you don't; it's that you may possess it to a different extent from others in the field. Although it's true that all three skills are fundamental and that you need to have each to a degree, you don't need to be an expert in all of them. We put within the triangle different types of data science specialties. These specialties don't always map one-to-one with job titles, and even when they do, different companies sometimes call them different things.

So what does each of these components mean?

1.1.1 Mathematics/statistics

At the basic level, mathematics and statistics knowledge is data literacy. We break down that literacy into three levels of knowledge:

- *That techniques exist*—If you don't know that something is possible, you can't use it. If a data scientist was trying to group similar customers, knowing that statistical methods (called *clustering*) can do this would be the first step.
- *How to apply the techniques*—Although a data scientist may know about many techniques, they also need to be able to understand the complexities of applying them—not only how to write code to apply the methods, but also how to configure them. If the data scientist wants to use a method such as *k*-means clustering to group the customers, they would need to understand how to do *k*-means

clustering in a programming language such as R or Python. They would also need to understand how to adjust the parameters of the method, for example, by choosing how many groups to create.

- *How to choose which techniques to try*—Because so many possible techniques can be used in data science, it's important for the data scientist to be able to quickly assess whether a technique would work well. In our customer grouping example, even after the data scientist focuses on clustering, they have to consider dozens of different methods and algorithms. Rather than trying each method, they need to be able to rule out methods quickly and focus on just a few.

These sorts of skills are used constantly within a data science role. To consider a different example, suppose that you work at an e-commerce company. Your business partner might be interested in what countries have the highest average order value. If you have the data available, this question is easy to answer. But rather than simply presenting this information and letting your customer draw their own conclusions, you could dig deeper. If you have one order from country A for $100, and a thousand orders from country B that average $75, it is correct that country A has the higher average order value. But would you be confident in saying that this means your business partner should definitely invest in advertising in country A to increase the number of orders? Probably not. You have only one data point for country A, and maybe it's an outlier. If country A had 500 orders instead, you might use a statistical test to see whether the order value was significantly different, meaning that if there really was no difference between A and B on this measure, you'd be unlikely to see the difference you did. In this one paragraph-long example, many different assessments were made on what approaches were sensible, what should be considered, and what results were deemed to be unimportant.

1.1.2 Databases/programming

Programming and databases refer to the ability to pull data from company databases and to write clean, efficient, maintainable code. These skills are in many ways similar to what a software developer has to know, except that data scientists have to write code that does open-ended analysis rather than produces a predefined output. Each company's data stack is unique, so no one set of technical skills is required for a data scientist. But broadly, you'll need to know how to get data from a database and how to clean, manipulate, summarize, visualize, and share data.

In most data science jobs, R or Python is the main language. R is a programming language that has its roots in statistics, so it's generally strongest for statistical analysis and modeling, visualization, and generating reports with results. Python is a programming language that started as a general software development language and has become extremely popular in data science. Python is known for being better than R at working with large datasets, doing machine learning, and powering real-time algorithms (such as Amazon's recommendation engines). But thanks to the work of many contributors, the two languages' capabilities are now at near parity. Data scientists are

successfully using R to make machine learning models that are run millions of times a week, and they're also making clean, presentable statistical analyses in Python.

R and Python are the most popular languages for data science for a couple of reasons:

- They're free and open source, meaning that many people, not just one company or one group, contribute code that you can use. They have many packages or *libraries* (sets of code) for doing data collection, manipulation, visualization, statistical analysis, and machine learning.
- Importantly, because each language has such a large following, it's easy for data scientists to find help when they run into problems. Although some companies still use SAS, SPSS, STATA, MATLAB, or other paid programs, many of them are starting to move to R or Python instead.

Although most data science analysis is done in R or Python, you'll often need to work with a database to get the data. This is where the language SQL comes in. SQL is the programming language that most databases use to manipulate data within them or to extract it. Consider a data scientist who wants to analyze the hundreds of millions of records of customer orders in a company to forecast how orders per day will change over time. First, they would likely write a SQL query to get the number of orders each day. Then they would take those daily order counts and run a statistical forecast in R or Python. For this reason, SQL is extremely popular in the data science community, and it's difficult to get too far without knowing it.

Another core skill is using *version control*—a method of keeping track of how code changes over time. Version control lets you store your files; revert them to a previous time; and see who changed what file, how, and when. This skill is extremely important for data science and software engineering because if someone accidentally changes a file that breaks your code, you want the ability to revert or see what changed.

Git is by far the most commonly used system for version control and is often used in conjunction with GitHub, a web-based hosting service for Git. Git allows you to save (*commit*) your changes, as well as see the whole history of the project and how it changed with each commit. If two people are working on the same file separately, Git makes sure that no one's work is ever accidentally deleted or overwritten. At many companies, especially those with strong engineering teams, you'll need to use Git if you want to share your code or put something into production.

Can you be a data scientist without programming?

It's possible to do a lot of data work using only Excel, Tableau, or other business intelligence tools that have graphical interfaces. Although you're not writing code, these tools claim to have much of the same functionality as languages such as R or Python, and many data scientist do use them sometimes. But can they be a complete data science toolkit? We say no. Practically, very few companies have a data science team where you wouldn't need to program. But even if that weren't the case, programming has advantages over using these tools.

The first advantage of programming is reproducibility. When you write code instead of using point-and-click software, you're able to rerun it whenever your data changes, whether that's every day or in six months. This advantage also ties into version control: instead of renaming your file every time your code changes, you can keep one file but see its entire history.

The second advantage is flexibility. If Tableau doesn't have a type of graph available, for example, you won't be able to create it. But with programming, you can write your own code to make something that the creators and maintainers of a tool never thought of.

The third and final advantage of open source languages such as Python and R is community contribution. Thousands of people create *packages*, and publish them openly on GitHub and/or CRAN (for R) and pip (for Python). You can download this code and use it for your own problems. You're not reliant on one company or group of people to add features.

1.1.3 Business understanding

Any sufficiently advanced technology is indistinguishable from magic.

Arthur C. Clarke

Businesses have, to put it mildly, varying understanding of how data science works. Often, management just wants something done and turns to its data science unicorns to make that thing happen. A core skill in data science is knowing how to translate a business situation into a data question, find the data answer, and finally deliver the business answer. A businessperson might ask, for example, "Why are our customers leaving?" But there's no "why-are-customers-leaving" Python package that you can import—it's up to you to deduce how to answer that question with data.

Business understanding is where your data science ideals meet the practicalities of the real world. It's not enough to want a specific piece of information without knowing how the data is stored and updated at your specific company. If your company is a subscription service, where does the data live? If someone changes their subscription, what happens? Does that subscriber's row get updated, or is another row added to the table? Do you need to work around any errors or inconsistencies in the data? If you don't know the answers to these questions, you won't be able to give an accurate answer to a basic question like "How many subscribers did we have on March 2, 2019?"

Business understanding also helps you know what questions to ask. Being asked "What should we do next?" by a stakeholder is a little like being asked "Why do we not have more money?" This type of question begs more questions. Developing an understanding of the core business (as well as the personalities involved) can help you parse the situation better. You might follow up with "Which product line are you looking for guidance regarding?" or "Would you like to see more participation from a certain sector of our audience?"

Another part of business understanding is developing general business skills, such as being able to tailor your presentations and reports to different audiences. Sometimes, you'll be discussing a better methodology with a room full of statistics PhDs, and sometimes, you'll be in front of a vice president who hasn't taken a math class in 20 years. You need to inform your audience without either talking down or overcomplicating.

Finally, as you become more senior, part of your job is to identify where the business could benefit from data science. If you've wanted to build a prediction system for your company but have never had management support, becoming part of the management team can help solve that problem. A senior data scientist will be on the lookout for places to implement machine learning, as they know its limitations and capabilities, as well as which kinds of tasks would benefit from automation.

Will data science disappear?

Underlying the question about whether data science will be around in a decade or two are two main concerns: that the job will become automated and that data science is overhyped and the job-market bubble will pop.

It's true that certain parts of the data science pipeline can be automated. Automated Machine Learning (AutoML) can compare the performance of different models and perform certain parts of data preparation (such as scaling variables). But these tasks are just a small part of the data science process. You'll often need to create the data yourself, for example; it's very rare to have perfectly clean data waiting for you. Also, creating the data usually involves talking with other people, such as user experience researchers or engineers, who will conduct the survey or log the user actions that can drive your analysis.

Regarding the possibility of a pop in a job-market bubble, a good comparison is software engineering in the 1980s. As computers grew cheaper, faster, and more common, there were concerns that soon a computer could do everything and that there would be no need for programmers. But the opposite thing happened, and now there are more than 1.2 million software engineers in the United States (http://mng.bz/MOPo). Although titles such as webmaster did disappear, more people than ever are working on website development, maintenance, and improvement.

We believe that there will be more specialization within data science, which may lead to the disappearance of the general title data scientist, but many companies are still in the early stages of learning how to leverage data science and there's plenty of work left to do.

1.2 Different types of data science jobs

You can mix and match the three core skills of data science (covered in section 1.1) into several jobs, all of which have some justification for having the title data scientist. From our perspective, these skills get mixed together in three main ways: analytics, machine learning, and decision science. Each of those areas serves a different purpose for the company and fundamentally delivers a different thing.

When looking for data science jobs, you should pay less attention to the job titles and more to the job descriptions and what you're asked in the interviews. Look at the backgrounds of people in data science roles, such as what previous jobs they held and what their degrees are. You may find that people who work in similar-sounding jobs have totally different titles or that people who have the same data scientist title do totally different things. As we talk in this book about different types of data science jobs, remember that the actual titles used at companies may vary.

1.2.1 Analytics

An *analyst* takes data and puts it in front of the right people. After a company sets its yearly goals, you might put those goals in a dashboard so that management can track progress every week. You could also build in features that allow managers to easily break down the numbers by country or product type. This work involves a lot of data cleaning and preparation but generally less work to interpret the data. Although you should be able to spot and fix data quality issues, the primary person who makes decisions with this data is the business partner. Thus, the job of an analyst is to take data from within the company, format and arrange it effectively, and deliver that data to others.

Because the analyst's role doesn't involve a lot of statistics and machine learning, some people and companies would consider this role to be outside the field of data science. But much of the work, such as devising meaningful visualizations and deciding on particular data transformations, requires the same skills used in the other types of data science roles. An analyst might be given a task such as "Create an automated dashboard that shows how our number of subscribers is changing over time and lets us filter the data to just subscribers of specific products or in specific geographical regions." The analyst would have to find the appropriate data within the company, figure out how to transform the data appropriately (such as by changing it from daily to weekly new subscriptions), and then create a meaningful set of dashboards that are visually compelling and automatically update each day without errors.

Short rule: an analyst creates *dashboards and reports that deliver data.*

1.2.2 Machine learning

A *machine learning engineer* develops machine learning models and puts them into production, where they run continuously. They may optimize the ranking algorithm for the search results of an e-commerce site, create a recommendation system, or monitor a model in production to make sure that its performance hasn't degraded since it was deployed. A machine learning engineer spends less time on things like creating visualizations that will convince people of something and more time doing the programming work of data science.

A big difference between this role and other types of data science positions is that the work output is primarily for machine consumption. You might create machine learning models that get turned into application programming interfaces (APIs) for

other machines, for example. In many ways, you'll be closer to a software developer than to other data science roles. Although it's good for any data scientist to follow best coding practices, as a machine learning engineer, you must do so. Your code must be performant, tested, and written so that other people will be able to work with it. For this reason, many machine learning engineers come from a computer science background.

In a machine learning engineer role, a person may be asked to create a machine learning model that can—in real time—predict the probability that a customer on the website will actually finish their order. The machine learning engineer would have to find historical data in the company, train a machine learning model on it, turn that model into an API, and then deploy the API so that the website can run the model. If that model stops working for some reason, the machine learning engineer will be called to fix it.

Short rule: a machine learning engineer creates *models that get run continuously*.

1.2.3 Decision science

A *decision scientist* turns a company's raw data into information that helps the company make decisions. This work relies on having deep understanding of different mathematical and statistical methods and familiarity with business decision-making. Furthermore, decision scientists have to be able to make compelling visualizations and tables so that the nontechnical people they talk to will understand their analysis. Although a decision scientist does plenty of programming, their work generally gets run only once to make a particular analysis, so they can get away with having code that's inefficient or difficult to maintain.

A decision scientist must understand the needs of the other people within the company and figure out how to generate constructive information. A marketing director, for example, might ask a decision scientist to help them decide which types of products should be highlighted in the company's holiday gift guide. The decision scientist might investigate what products have sold well without being featured in the gift guide, talk to the user research team about conducting a survey, and use principles of behavioral science to do an analysis to come up with the optimal items to suggest. The result is likely to be a PowerPoint presentation or report to be shared with product managers, vice presidents, and other businesspeople.

A decision scientist often uses their knowledge of statistics to help the company make decisions under uncertainty. A decision scientist could be responsible for running their company's experimentation analytics system, for example. Many companies run online experiments, or A/B tests, to measure whether a change is effective. This change could be as simple as adding a new button or as complicated as changing the ranking system of search results or completely redesigning a page. During an A/B test, visitors are randomly assigned to one of two or more conditions, such as half to the old version of the home page, which is the *control*, and half to the new version, which is the *treatment*. Then visitors' actions after they enter the experiment are compared

to see whether those in the treatment have a higher rate of performing desirable actions, such as buying products.

Because of randomness, it's rare for the metrics in the control and treatment to be exactly the same. Suppose that you flipped two coins, and that one turned up heads 52 times out of 100 and one 49 times out of 100. Would you conclude that the first coin is more likely to turn up heads? Of course not! But a business partner might look at an experiment, see that the conversion rate is 5.4 percent in the control and 5.6 percent in the treatment, and declare the treatment to be a success. The decision scientist is there to help interpret the data, enforce best practices for designing experiments, and more.

Short rule: a decision scientist creates analyses that produce *recommendations*.

1.2.4 Related jobs

Although the three areas discussed in the preceding sections are the main types of data science positions, you may see a few other distinct roles that fall outside those categories. We list those jobs here, because it's good to understand the positions that are out there and because you may need to collaborate with colleagues in these positions. That said, if you're interested in one of these roles, the material in this book may be less relevant to you.

Business intelligence analyst

A *business intelligence analyst* does work similar to that of an analyst, but they generally use less statistical and programming expertise. Their tool of choice may be Excel instead of Python, and they may not ever make statistical models. Although their job function is similar to that of an analyst, they create less-sophisticated output because of the limitations of their tools and techniques.

If you want to do machine learning or programming, or to apply statistical methods, a business intelligence analyst position could be a very frustrating role, because it won't help you gain new skills. Also, these jobs usually pay less than data science jobs and are considered to be less prestigious. But a business intelligence analyst job can be a good entry point to becoming a data scientist, especially if you haven't worked with data in a business setting before. If you want to start as a business intelligence analyst and grow into becoming a data scientist, look for positions in which you can learn some skills you may not have, such as programming in R or Python.

Data engineer

A *data engineer* focuses on keeping data maintained in databases and ensuring that people can get the data they need. They don't run reports, make analyses, or develop models; instead, they keep the data neatly stored and formatted in well-structured databases so that other people can do those things. A data engineer may be tasked with maintaining all the customer records in a large-scale cloud database and adding new tables to that database as requested.

Data engineers are pretty different from data scientists, and they're even more rare and in demand. A data engineer may help build the data backend components of a

company's internal experimentation system and update the data processing flow when the jobs start taking too long. Other data engineers develop and monitor batch and streaming environments, managing data from collection to processing to data storage.

If you're interested in data engineering, you'll need strong computer science skills; many data engineers are former software engineers.

RESEARCH SCIENTIST

A *research scientist* develops and implements new tools, algorithms, and methodologies, often to be used by other data scientists within the company. These types of positions almost always require PhDs, usually in computer science, statistics, quantitative social science, or a related field. Research scientists may spend weeks researching and trying out methods to increase the power of online experiments, getting 1% more accuracy on image recognition in self-driving cars, or building a new deep learning algorithm. They may even spend time writing research papers that may rarely be used within the company but that help raise the prestige of the company and (ideally) advance the field. Because these positions require very specific backgrounds, we don't focus on them in this book.