IN THIS CHAPTER

- Understanding what data science is and why it matters
- Seeing how data science, big data, and AI connect
- Following the data science pipeline from raw data to insight
- Meeting the Python libraries that power data science
- Installing and verifying your data science dependencies

Chapter 1
What Is Data Science?

Errors using inadequate data are much less than those using no data at all.

—CHARLES BABBAGE

If you've read Book 2, you already know Python. You can write functions, work with lists and dictionaries, loop through data, and pull in external libraries. That's a solid base, and it's just what you need to enter one of today's most sought-after tech careers: data science.

This minibook puts those Python skills to work on real problems. You load datasets, clean messy data, visualize patterns, and build your first predictive model. By the time you're done, you'll have a working understanding of the full data science workflow and the tools professionals use every day.

But before diving into code, it's worth taking a chapter to understand what data science is, how it works as a process, and which Python libraries you'll be reaching for throughout this minibook. That context will make everything that follows click faster.

Defining Data Science

You might think that understanding data science involves complex equations and requires an advanced degree. In practice, though, data science is something far more accessible: It's the process of extracting useful insights from data. This process might mean figuring out why sales dropped last quarter, predicting which customers are likely to cancel their subscriptions, identifying fraudulent transactions in real time, or helping a streaming service recommend what to watch next.

How data science got its name

The term *data science* is surprisingly recent. William S. Cleveland coined it in 2001 in a paper titled "Data Science: An Action Plan for Expanding the Technical Areas of the Field of Statistics." The International Council for Science formally recognized the field a year later, and Columbia University began publishing the *Journal of Data Science* in 2003.

The name is new, but the core concepts are very old. For thousands of years, people have been using statistics, which is the mathematical foundation of data science, to identify patterns in data. For example, the historian Thucydides describes Athenian soldiers in the fifth century BC carefully counting bricks in an unplastered section of a city wall and then averaging those counts across multiple soldiers to estimate the wall's full height. This is an early example of data collection, reducing errors, and making inferences, which are the essential parts of data science that we still practice 2,500 years later.

What's new isn't the math. It's the scale of the data, the power of the computers, and the libraries available that make the process practical for anyone who can write a few lines of Python.

What data scientists do

Data scientists rarely work alone, and for good reason: the job requires a wide range of skills. In practice, teams often divide the work across three broad competency areas:

- **Data capture:** Obtaining data by connecting to databases, pulling information from APIs, and loading files. This process also requires a strong understanding of the domain to identify which questions are valuable to explore.
- **Analysis:** Applying statistical methods and algorithms to find patterns, test hypotheses, and draw conclusions that wouldn't be visible from simply reading the raw numbers.

» **Presentation:** Translating findings into visualizations and plain language so that people who weren't part of the analysis can understand and act on the results.

Over the course of this minibook, you touch on all three areas.

Data science, big data, and AI

Data science is not an isolated discipline; it combines statistics, software development, and domain expertise, and it directly supports the fields of machine learning and artificial intelligence.

You may have heard the term *ETL* (extract, transform, and load), which describes the process of pulling data from multiple sources, converting it into a consistent format, and loading it somewhere useful for analysis. You follow this process in Chapter 3 of this minibook when you load your first datasets.

The outputs of data science analysis often feed directly into AI applications. When a music service learns your listening preferences and starts suggesting artists you've never heard of but end up loving, that's data science at work upstream of an AI recommendation system. When a hospital system flags a patient's test results as potentially indicating a condition before symptoms appear, that's data science informing a clinical AI tool. The pipeline from raw data to insight to automated action is one of the most consequential technology systems of our time.

The Data Science Pipeline

Data science is partly art and partly engineering. The art is in knowing which questions to ask, which patterns to look for, and how to present findings compellingly. The engineering is in the repeatable set of steps that transforms raw, messy data into a reliable, interpretable result.

This process is called the *data science pipeline,* and understanding it gives you a map for everything else in this minibook.

Step 1: Preparing the data

Real-world data is rarely clean. It arrives from multiple sources in different formats, and with missing values, duplicate records, inconsistent naming conventions, and all manner of other problems. Before you can analyze data, you have to

prepare it, which involves loading it, examining it, cleaning it, and transforming it into a form your tools can work with.

Many experienced data scientists spend 70 to 80 percent of their time on data preparation. Chapters 2 and 3 of this minibook focus on data preparation.

Step 2: Exploring the data

Once your data is in reasonable shape, you explore it. *EDA*, or *exploratory data analysis*, involves calculating descriptive statistics, looking at distributions, spotting outliers, and visualizing relationships between variables before committing to any particular modeling approach.

EDA is where you ask: What does this data look like? Are there patterns? Anything surprising? Anything that seems wrong? It's an iterative process in which you form a hypothesis, look at the data, revise your hypothesis, and look again. Chapter 4 of this minibook covers the EDA tools you'll use most: descriptive statistics and visualization.

Step 3: Learning from the data

With a solid understanding of the data, you can apply algorithms to find deeper patterns and make predictions. This is the machine learning part of data science, in which you train a model on your data so that it can generalize to new examples. Chapter 5 walks you through this process.

Step 4: Visualizing

Numbers tell a story, but most people need a picture to see it. Visualization is woven throughout the pipeline, from EDA to model evaluation. A well-constructed chart can reveal in seconds what a table of numbers might obscure completely.

Step 5: Obtaining insights and data products

The pipeline doesn't end until you've extracted something useful such as a decision, a recommendation, a prediction, or a report. The goal is never to produce a model; it's to answer a question or solve a problem. A data product is what gets deployed. Examples of data products include a pricing algorithm, a spam filter, or a dashboard that updates in real time.

The Python Data Science Ecosystem

You may have started learning Python for general programming, or maybe data science was the goal from the start. Either way, you've chosen the right language. Python is the dominant tool in data science for a straightforward reason: Its ecosystem of scientific libraries is unmatched.

Other languages can perform data science tasks. R has deep statistical roots. Julia is fast. MATLAB is widely used in engineering and research. But Python is unique in that it's simultaneously a powerful general-purpose language *and* the best-supported environment for data science. You can write the data pipeline, build the model, create the API that serves predictions, and build the web interface that displays them — all in the same language.

Python's power in data science comes from a collection of specialized libraries you import into your programs. You've already used libraries in Book 2 (the statistics module, datetime, and others). The data science libraries work the same way, just at a much larger scale. This section introduces the main libraries you'll encounter in this minibook.

NumPy

NumPy (`numpy.org`) is the foundation of scientific computing in Python. Its central feature is the *n-dimensional array* (or *ndarray*), which is an efficient and fast data structure for storing and operating on large collections of numbers. Whereas a Python list is flexible but relatively slow when you're doing math on thousands of values, a NumPy array is designed for that kind of work.

NumPy also provides functions for linear algebra, random-number generation, and mathematical operations that can be applied to entire arrays at once. Almost every other data science library is built on top of NumPy.

pandas

pandas (`pandas.pydata.org`) builds on NumPy to provide data structures and analysis tools designed for real-world data. Its central object is the *DataFrame*, which is like a spreadsheet that you can manipulate with Python code. You can load a CSV file into a DataFrame with one line of code. DataFrames make it simple to filter rows, calculate column statistics, handle missing values, merge datasets, and export the result.

If NumPy is the engine, pandas is the dashboard. It's the library you'll use most for day-to-day data work.

Matplotlib

Matplotlib (`matplotlib.org`) is Python's foundational plotting library. It gives you precise control over every element of a chart, including axes, labels, colors, line styles, and legends. With Matplotlib, you can produce publication-quality figures. You'll use Matplotlib in Chapter 4 to visualize your data during exploratory analysis and to communicate findings in Chapter 5.

Scikit-learn

Scikit-learn (`scikit-learn.org`) is the go-to library for machine learning in Python. It provides a consistent, well-designed interface for dozens of algorithms along with tools for evaluating model performance and preparing data. Chapter 5 introduces Scikit-learn and walks you through training your first predictive model.

SciPy

SciPy (`scipy.org`) provides additional scientific computing tools built on NumPy: optimization, integration, signal processing, statistics, and more. You won't use SciPy directly much in this minibook, but it works in the background as part of the ecosystem, and you'll encounter references to it as you go deeper into data science.

REMEMBER

You don't need to master all these libraries at once. This minibook focuses on the ones you'll use every day and introduces them in context, as you need them. By the end of Chapter 5, each of them will feel familiar.

TIP

All the code in this minibook is written for Jupyter Notebook, which you're already familiar with from Book 2. Open a new notebook, and you're ready to go.

Installing the Libraries

Before you move on to Chapter 2, let's make sure you're all set to run Python code. If you followed the steps to install Python, VS Code, and get started with Jupyter Notebook in Book 2, Chapter 1, you're all set. If you haven't set up Python yet,

or if it's been a while, visit Chapter 1 of Book 2 now and come back here when you're done.

Do the following to install the libraries:

1. **Open VS Code and download the code for this book at `www.dummies.com/go/codingaiofd3e`.**
2. **Open the Book3 folder, and then open the Chapter 1 file (`chapter_01_what_is_data_science.ipynb`).**
3. **In the Chapter 1 file, find and run the two code cells by clicking the arrow to their left (the execute cell icon).**

   The first code cell installs the libraries you'll learn about in the rest of this minibook. The second one confirms that the libraries are installed and that you're ready to move on to Chapter 2.

   When everything is ready and installed, you'll see the screen shown in Figure 1-1.

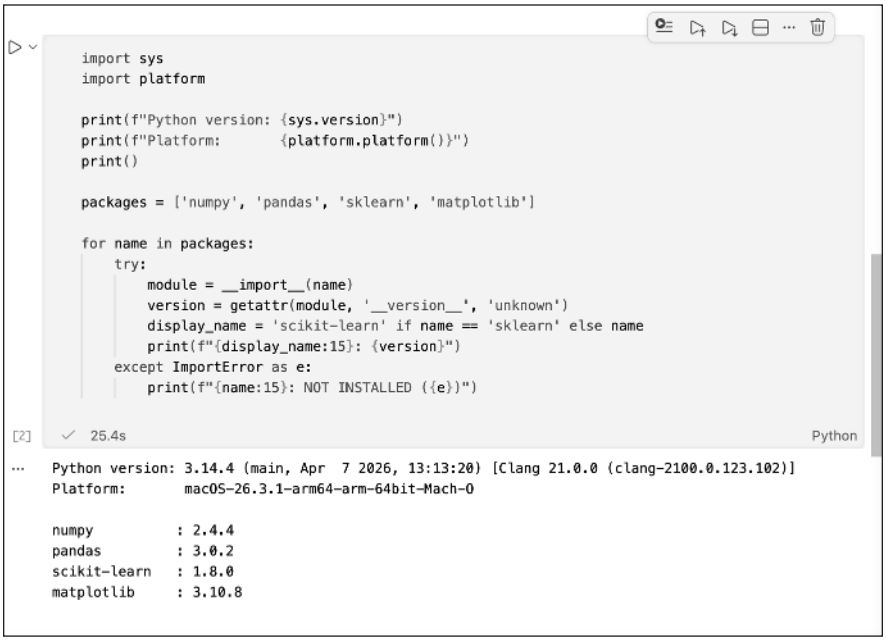

FIGURE 1-1: The libraries are installed!

If you encounter any issues while installing the libraries or running the demo, remember that you can ask your favorite AI assistant for help. Or if you don't mind waiting a bit longer, feel free to email me, and I'll do my best to help. (You'll find my email address in the Introduction.)