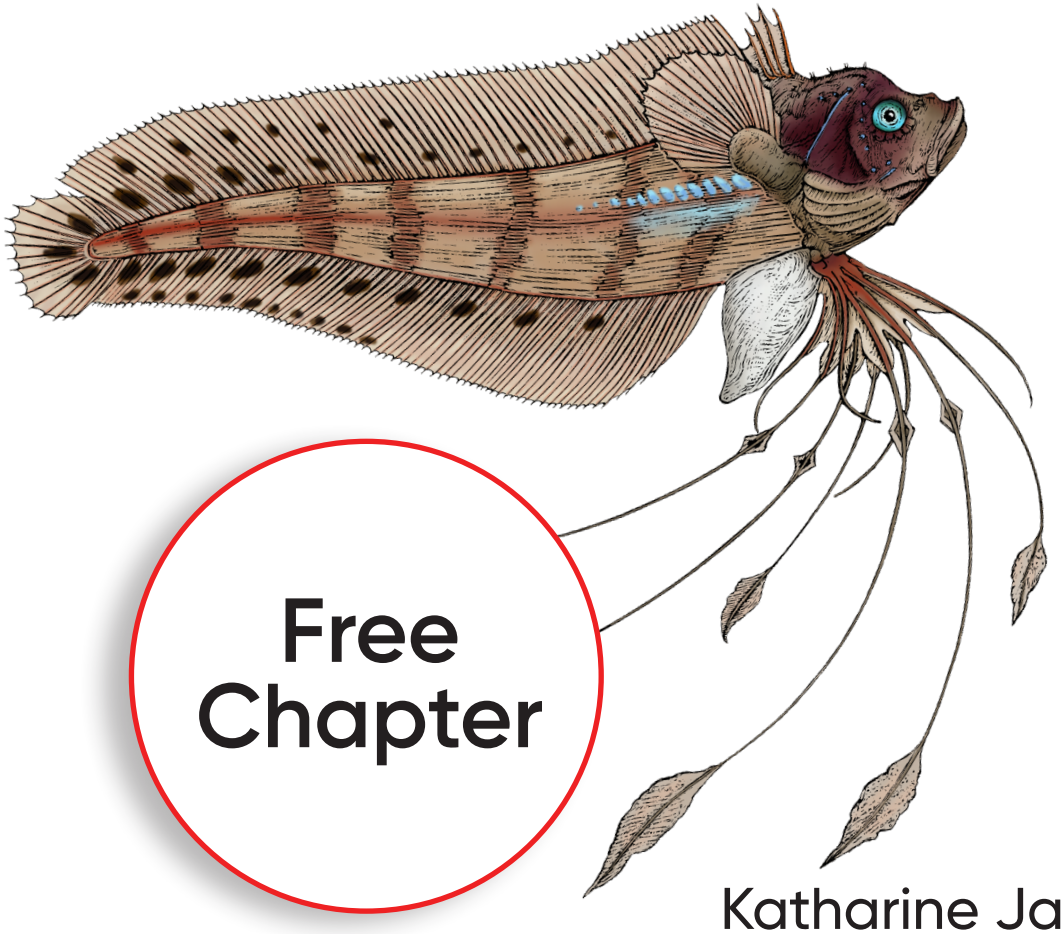
O'REILLY®

Practical Data Privacy

Enhancing Privacy and Security in Data



Free Chapter

Katharine Jarmul
Foreword by
Dr. Nakeema Damali Stefflbauer

Practical Data Privacy

Enhancing Privacy and Security in Data

This excerpt contains Chapter 1. The complete book is available on the O'Reilly Online Learning Platform and through other retailers.

Katharine Jarmul

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Practical Data Privacy
by Katharine Jarmul

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Andy Kwan
Development Editor: Rita Fernando
Production Editor: Kristen Brown
Copyeditor: Kim Wimpsett
Proofreader: Piper Editorial Consulting, LLC
Indexer: WordCo Indexing Services, Inc.
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Kate Dullea

April 2023: First Edition

Revision History for the First Edition
2023-04-19: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781098129460* for release details.

978-1-098-12946-0

[LSI]

Table of Contents

CHAPTER 1

Data Governance and Simple Privacy Approaches

Data privacy is a large and long-lived field. I want you to picture it like an old road, packed with interesting side streets and diversions but hard to navigate if you don't know the way. This chapter is your initial orientation to this road. In this chapter and throughout this book, I'll help you map important parts of the privacy landscape, and you'll find areas where you want to learn more and deviate from the original path. Applying this map within your organization means uncovering who is doing what, what their responsibilities are, and what data privacy needs exist in your organization.[1]

You might have heard the phrase *data governance* only once or hundreds of times, but it is often left unexplained or open for interpretation. In this chapter, you'll learn where data governance overlaps with data privacy for practical data science purposes and learn simpler approaches for solving privacy problems with data, such as pseudonymization. You'll also learn how governance techniques like documentation and lineage tracking can help identify privacy problems or ways to implement privacy techniques at the appropriate step.

1 Throughout this book, I'll use the term *organization* as a word to describe your workplace. If you are at a small agile data science consultancy, a massive corporation, or a midsize nonprofit, you will have a vastly different experience. This book should be useful for all groups—take the advice and learnings and use your own knowledge of your work to fit them to your size and culture.

If you already know or work in data governance, I recommend skimming or skipping this chapter. If governance and data management are new to you, this chapter will show you the foundations needed to apply the advanced techniques you'll learn in later chapters.

This chapter will help give you tools and systems to identify, track, and manage sensitive data. Without this foundation, it will be difficult to assess privacy risk and mitigate those concerns. Starting with governance makes sense, because privacy fits well into the governance frameworks and paradigms, and these areas of work support one another in data systems.

Data Governance: What Is It?

Data governance is often used as an "all-encompassing" way to think about our data decisions, like whether to opt in to allowing a service to contact you or determining who has access rights to a given database. But what does the phrase really refer to, and how can you make it actionable?

Data governance is literally governing data. One way to govern happens via a transfer of rights people individually and communally possess. Those rights are passed onto elected officials who manage tasks and responsibilities for individuals who have no time, expertise, or interest. In data governance, individuals transfer rights when data is given to an organization. When you use a website, service, or application, you agree to whatever privacy policy, terms, and conditions or contract is presented by those data processors or collectors at that time. This is similar to living in a particular state and implicitly agreeing to follow the laws of that land.

Data governance helps manage whose data you collect, how you collect and enhance it, and what you do with it after collection. Figure 1-1 illustrates how privacy and security relate to data governance, via an imaginary island where users and their data are properly protected by both privacy and security initiatives. In this diagram, you can see the sensitive data inside a tower. Security initiatives are supported by Privacy by Design.[2] Regulations and compliance provide a moat that keeps sensitive data separate. Privacy technologies you will learn in this book are bridges for users and data stakeholders, allowing them to gather insights and make decisions with sensitive data without violating individual privacy.

2 Privacy by Design is a set of principles developed by Ann Cavoukian outlining measures technologists can use to ensure systems are architected and software is designed with privacy in mind from the beginning. You will hear it used quite often in conversations with experienced governance experts. I recommend taking time to read and explore these principles and determine how they fit your data work. These principles are included in Chapter 11.

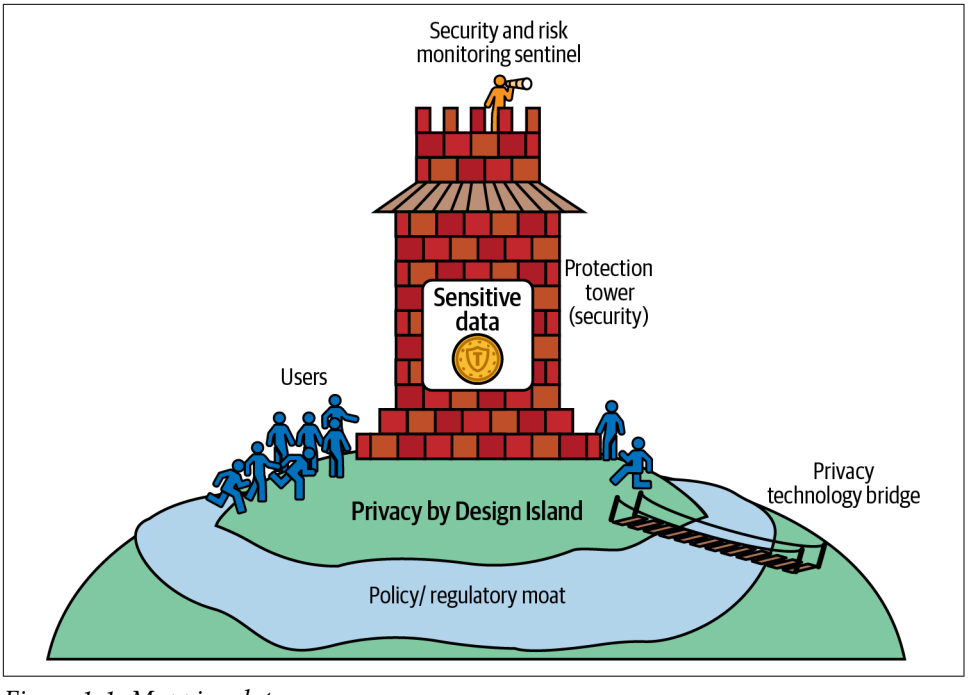


Figure 1-1. Mapping data governance

Data governance can be described as a mixture of people, process, and technology. Regardless of your organization size, there is always some amount of data governance work to be done. If you are at a large organization, there is probably a large team or committee creating standards, which turn into policies and procedures; those then need to be implemented in the organization's technology. If your organization is small, this might be the job of your technical or legal leader. Let's zoom into the technology section, as this is likely where you'll be asked to help take these policies and procedures and ensure they are actually part of regular data processing.

What elements of taking governance standards and policies and implementing them in technology are important for data scientists? Table 1-1 outlines significant areas and related questions within data governance that you will grapple with as a data scientist.

Table 1-1. Data governance in data science

Data lineage/Origin	Policies and controls
Where did the data come from?	What laws or internal policies apply to this data?
Whose data is it? Is it possible to contact them?	Where, when, and how was it collected?
Was this data acquired from someone else, and if so, did they document how it was processed and who it belongs to?	What privacy or security concerns do you need to address when using this data?
How did the processing change the data?	What was the privacy policy and terms at collection time?
Is the metadata for lineage information easily accessible and queryable?	Did the data come from a third party? If so, what are the restrictions and obligations, contractual or otherwise, for this data?

Data reliability/Knowledge	Data privacy and security
What are the concerns around understanding the data and systems (i.e., including collection, transformation, and downstream systems)?	How is access to sensitive data managed and monitored?
Does the data have an understandable documentation trail from the moment it was collected?	Does the organization know if and when data has been breached? How?
When there are data quality problems, do you know how to diagnose and resolve them?	Who is responsible for managing privacy controls? Security controls?
Are there data storage infrastructure or old data stores that are undocumented or even unknown?	When someone invokes their data rights (i.e., GDPR), is there a system that is well documented and understood to apply these rights?
Is the data well documented and understood? (Know your data.)	What data loss prevention technologies and privacy technologies do you use and how?

You are likely already focused on many of these questions since data is a major part of your job. You might have personally suffered from a lack of data documentation, incomplete understanding of how a certain database came to be, and issues with data labeling and quality. Now you have a new word to use to describe these qualities: *governance*!

Working on the governance side of data administration or management is really about focusing on how to collect and update information about the data throughout its lifecycle. The regulatory, privacy, and security concerns shape that information and ensure governance decisions and frameworks expedite measures like individual data rights and appropriate usage of data. If your data does not come from individuals, there may be other concerns with regard to proprietary data or related security issues that guide governance initiatives.

When you think about governing data in a concrete way, you begin to look at tasks such as documenting the ever-changing data flows at your organization. It seems obvious and easy, but on closer look it is anything but.

Let's say you have a huge data lake that gets fed from 10 different sources, some external, some internal. How can you actually begin to govern that data? What would a scalable and easy-to-use solution look like? What happens when those data flows change? It may be enough just to document the code or the workflows that are actively running and in use and to leave the rest for future work. But what do you do with data from partners or other external data collection systems? You'll need to coordinate this documentation so the legal, privacy, and risk departments can use it for auditing and assessment. This process should not be solved with piecemeal and temporary solutions but instead addressed as holistically as possible.

To begin, let's identify which data is the most important to protect for the purpose of practical data privacy. How can you identify sensitive data? What exactly is sensitive data?

Identifying Sensitive Data

In the context of privacy, sensitive data is normally defined as person-related data or even just personally identifiable data.[3] It includes your full name, your email address, your gender, your mailing address, your IP address, your social media profile, your phone number, your Social Security number or other national identification number, your credit card number, your birthday, your health records or biometric data (i.e., fingerprints, iris scan, or even your gait!).

All of these fall under the category of personally identifiable data, or what many refer to as *personally identifiable information* (PII). This data is specific to you; it can be used alone or in combination with other sources of information about you to directly identify, indirectly identify, or re-identify you. This is the most sensitive and frequently regulated data because it provides a nearly or completely unique identifier.

As defined in this book, sensitive data includes:

PII
: Data that is unique or close to unique for a particular individual. This is usually defined in policy and regulations and can include things that you might not expect, such as your IP address, date of birth, and workplace.

Person-related data
: Data that relates to a person but doesn't fall under PII. This could be anything related to their personhood, including interests, beliefs, locations, and online and offline behaviors and activities.

3 Your organization might have their own definition of sensitive data that varies from the one used in this book. Ensure you use appropriate terms internally when bringing up these topics.

Proprietary and confidential data

: Data deemed sensitive for contractual or business-related purposes. Its release would endanger a business or other legal relationship or agreement.

One thing I hope this book will do is expand your definition of sensitive data to include a broader range. For example, do you think your phone location is sensitive? Or does it depend on where you are? If you are sitting at home, your phone location also reveals your mailing address or residence, which again is personally identifiable. What about if you are at work? Or in a movie theater? Or at a close friend's house?

What about your political affiliation and areas of interest? Voting history? Religious beliefs and practices? Your friendships, partnerships, and who you connect with? What about your daily routines, the news you consume, your music and entertainment choices, the devices you own?

These questions begin to reveal the range of privacy preferences. Some individuals might be comfortable sharing their location at work or might even be required to do so. Others might see this as an invasion of privacy. Whereas one person might be quite open about their personal relationships, political choices, and religion, another person might see these topics as deeply personal and sensitive. This echoes the notion of contextual privacy and social privacy as discussed in the Preface. Here, regulations step in, giving individuals more choice regarding their privacy preferences and the ability to communicate that sensitivity to data collectors via changes in privacy policies and consent choices.

It also means, however, that when you work with data, you recognize the range of what is considered *sensitive*. You should be aware of the additional privacy risk created when person-related data is combined in a new way that inadvertently exposes the individuals. For example, if I track your location throughout the day, I would likely learn where you work, where you eat, your daily personal activities, and where you live. Even if I were to collect data just while you were moving (i.e., driving data), I would be able to identify some of those attributes. Even if I collected data only when you were in the presence of others, I could likely still infer things about you, such as if you travel with family or a friend or if you like to shop at a particular store or commute along a particular route and at what times.

Similarly, it's been shown by researchers that a series of Facebook likes can be used to infer things such as gender, sexual orientation, and political beliefs—even the marital status of your parents.[4] These are inferences, not necessarily fact—but it is clear that online behavior and social network behavior create unique breadcrumbs and reveal patterns that expose personal and private traits of an individual. This proves any person-generated data is potentially identifiable.

The power of inference combined with vast quantities of information can identify individuals even when that is not the intention. In targeted advertising, sensitive attributes are often inferred and combined without consent, leading to recommendations that might leak sensitive information such as sexual orientation or political views. When an advertiser is choosing targeted groups, the more factors they specify, the more easily they could erroneously target a particular individual or very small target group. If an advertiser is acting maliciously, they can use the information they have to figure out how to target that exact person or get fairly close.

For these reasons, the term *sensitive data* could mean any person-related data, regardless of if it is directly personally identifiable or not. Person-related data, particularly in large quantities or in aggregate, is identifiable. In this book, when I say sensitive data, I am referring not only to PII but also to a broader range of person-related data, which could be used in combination with other information to identify a person or small group of people.

A final category of sensitive data is data that is proprietary or confidential for non-person-related reasons. This can be trade secrets, proprietary information about the business, or a particular product or information that falls under confidentiality clauses. This could be data shared between parent companies and their subsidiaries, which must be kept secret due to internal policies or confidentiality agreements. Or it could be sensitive internal data that, if leaked, would give competitors an edge or compromise the company in another manner. This type of sensitive information also benefits from approaches and technologies you'll learn in this book.

I am a supporter of whistleblowing. If you have data you believe should be publicly known but is considered sensitive, think about the techniques you will learn in this book as ways to release that data publicly or to the appropriate authorities in a responsible and thoughtful way.

4 This work was published by some of the researchers who later worked on Cambridge Analytica. See: Kosinki et al., "Private traits and attributes are predictable from digital records of human behavior", 2013.

The first step toward protecting sensitive data is reliably identifying it. Once data is identified and documented as sensitive, you can then figure out how best to protect it.

Identifying PII

PII falls under a particular legal category in most data protection regulations, which requires close attention when data governance is implemented at an organization. If your organization collects any personal data—even for employees—then this data often has special governance requirements. Frequently, there is a lack of documentation or categorization of PII because it often shows up in text files, log files, or other unstructured data, which are all notoriously poorly documented.

There are several tools built explicitly for PII discovery in unstructured data using a variety of methods. I've also seen teams successfully build their own tools and systems for PII discovery. Many tools use fairly brittle methods like regular expressions (which are strings used to match patterns) or string entropy (for finding things like application programming interface [API] keys, cryptographic keys, or passwords). I have also successfully built deep learning models to identify PII in message text. Your results with these approaches will vary and should be evaluated depending on your use cases.

PII discovery is never perfect and never will be. It's important to talk with your risk teams (privacy, legal, security) about this fact, whether you purchase a PII discovery toolkit or build your own. It's safest to treat human-input data as extremely sensitive (e.g., as PII), regardless of the "cleaning processes." If you would like to use human-input data without extra protections associated with PII, you need to properly identify risks and ensure they are addressed and appropriately tested.

If you are working with a backlog of undocumented data and you fear there is a lot of PII contained therein, take a look at an easy-to-use open source tool. After you see how far that will take you, determine whether you need to invest in a more advanced or expensive approach. I can recommend Microsoft's Presidio, which also includes some basic pseudonymization techniques covered later in this chapter.

The best approach to manage and track PII is to actually track the data as it comes in and to label and manage this data as it traverses the system so you don't need to aggressively search or discover it later. One of the ways you can start the habit and culture of detecting PII early and often is to build a culture of documentation around data collection and data use. Table 1-1 gives you a good start. To develop a comprehensive approach, you might involve numerous parts of your organization, including

the security team, the information and data stakeholders, and the infrastructure and IT departments.

Documenting Data for Use

Documenting your data—sensitive or otherwise—is an essential part of data governance. At larger organizations, you may also have a classification system for your data: where particular policies are applied to different categories or classes of data. For example, you might need to tag and label PII so that you ensure access to it is restricted. In this case, you can use privacy classification as an initial step for your documentation.

But documentation goes way beyond sensitive data categories. In this section, I'll walk you through ways to think about data documentation, including how to begin basic data documentation; how to find undocumented data; how to add lineage, collection, and processing information; and finally when to implement data version control. Many of these systems are necessary to create the baseline for how sensitive data is used and managed around the organization—allowing you to identify use cases that lend themselves to advanced privacy technologies you'll learn later in this book.

Basic Data Documentation

Data needs to be documented based on how it will be consumed. When you document data, think of your readers, as you would if you were documenting code. What is this reader going to understand? How will they find and access the documentation in their normal workflow? How will they search through the documentation? What words will they use? What is the most important information? How can you make it concise and helpful enough so they will actually read it? How do you make data more self-documenting and easy to update?

Although the topic of data documentation is not new, it's also not a widely used practice, especially within data science teams. Data science practices have shifted from research-oriented teams and analysis-driven dashboards to experimentation, failures, agile development, and deployment standards. This makes it even more important that others understand what is happening in data workflows and experiments via well-written documentation.

Like data and experiment version control, documenting data enables other teams to discover and utilize data sources that might have been obtuse or hard to find if not properly documented. In many organizations, data sources are often split or duplicated across teams in different parts of the organization. It can be challenging to get even the basic access and interoperability right, and producing documentation can often fall by the wayside.

It doesn't have to be that way if documentation is seen as an essential part of data work. Here are some ways to convince data management or business units that data documentation is worth the extra time and effort. Data documentation:

- Speeds up data experiments, which can lead to new data-driven insights and discoveries
- Enables cross-department and team collaboration
- Accelerates data access and use for all stakeholders
- Helps eliminate unknown or undocumented data
- Empowers data teams when deciding which data to use for new ventures, products, and models
- Signals to product teams what data is available for new ideas
- Shows analysts disparate datasets that could be used for new insights and reports
- Gives compliance and audit teams proper oversight and assists in new data security and privacy controls
- Reduces compliance and data security risk

Data governance within your organization will work only if there is functional and effective data documentation. Proper documentation can even integrate into identity management and access systems to give data administrators, data owners, and data managers easy ways to grant and revoke access based on documentation.

To empower a responsible AI-driven organization, you'll need some extra documentation to communicate any stereotypes or known biases in the data itself. In this case, I recommend taking time to read about data cards, which is a way to document data for consumption by nontechnical users. If your team is focused on consumer-facing machine learning systems, I recommend using both data cards and model cards to ensure you are providing accurate, fair, and reliable systems!

The documentation can collect information on any or all of the following sections and can expand beyond what is discussed in this chapter. If you are setting up new documentation, figure out what will work best given your constraints. Prioritize the most valuable sections for your organization and expand from there.

Data Collection

Explanation and related questions

Who, where, when, why, and how was data collected? A description of when each dataset was collected, what team or software managed the collection, any post-collection processing that was performed, and why the collection happened (i.e., under what circumstances and legality).

Example

A diagram showing the data flow, with documentation on when it was implemented, by whom, and a data snapshot of the data at that point in time. The diagram should also contain consent information for legal, privacy, and compliance stakeholders—or other legitimate interest reasons for data collection, should no consent be given.

Data Quality

Explanation and related questions

What standardization has the data undergone (if any)? What quality controls were established? How many null values or extreme values are present in the data? Has the data been checked or processed for duplicates or inconsistencies? Here you can also document schema or unit changes.

Example

An analysis of the data quality in a particular time frame, including the frequency of null values, the standardization and harmonization of values (i.e., values that are changed to percentages or that are converted to a standard unit), and the data bias and variance. A deeper investigation could show histograms across numerous dimensions, show covariance or correlation between attributes, and provide information on outliers or extreme values. Should the data shift significantly in a short period of time, you could employ monitoring and alerting.

Each of these aspects needs to be taken care of individually and then also as a complete picture. Applying only data quality tests without data privacy or security can end up inadvertently exposing sensitive data. Ensure you apply these governance mechanisms holistically and integrate the different measures thoroughly and completely.

Data Security

Explanation and related questions

What level of security risk applies to this data? What measures should be taken in use or access of this data? Perform a risk analysis on the sensitivity of the data and its infrastructure, architecture, and storage details. Provide information to

help other teams (i.e., security and operations) accurately model and assess risk and make decisions on access restrictions. Here you can also document security and privacy technologies used to mitigate risk. These mitigations should also be documented so data consumers—people who will use, analyze, or further process the data—know how to work with the data intelligently. You'll learn more about how to assess, mitigate, and document security risk in Chapter 4.

Example

An evaluation of how a particular mitigation reduces the data security risk and a recommendation should this mitigation successfully address assessed risk. Implementation details should be included along with the required decision records if this mitigation is implemented.

Data Privacy

Explanation and related questions

Does the data contain person-related information? If so, what Privacy by Design for legal or data protection requirements have been performed to ensure the data is properly handled? If PII or person-related data is stored, the documentation should include information about the jurisdiction of the data and what privacy policies and consent options were shared with the users at collection time. There should also be clear documentation on all privacy-preserving mechanisms performed with linked code and even commit hashes if possible. Ensure all person-related data is treated as sensitive data and that protection measures are adequately documented.

Example

A list of columns that contain person-related data along with a timestamp column declaring when data should be deleted based on the policy and jurisdiction when data was collected. Even better if the deletion has been automated.

Data Definitions

Explanation and related questions

What is the organization of the data? If tabular, what do the column names mean? What data types are there? What units of measurement are used? Clarify the meaning of jargon or codes in the data (i.e., shorthand or internal mappings). A description of the data fields, column names (if any), keys and values, codes, units of measure, and other details that help a new consumer of the data understand what they are using. If particular formatting standards are chosen, such as ISO date representations, 24-hour time, or other standard units, please include details on how this processing is done in case others need to find potential errors or if processing changes.

Example

A queryable and easily accessible list of columns, their descriptions, and data types including categorical column codes listed in a searchable tabular format for easy reference.

Descriptive Statistics

Explanation and related questions

What are the standard descriptive statistics of the dataset, such as variance, distribution, mean, and so forth? How is the data distributed across particular features of importance? Is there concern that the data has particular biases or unequal classes? Summarized statistical descriptions of the data can be included in written or graphical form to allow others to quickly assess if the data fits their needs. Here, an interactive experience goes a long way. These properties can be quite sensitive, so you should expose them only after assessing the security and privacy risks of granting access. One great tool to look at for inspiration is Google's Facets.

Example

A chart showing the percentiles of numerical columns (using box charts with outliers removed, for example). These could be selectable and dynamic based on other features, which allows you to analyze correlations and biases in the dataset itself.

This list is not all-encompassing, but it can guide you as you begin to address data documentation at your organization. Remember, this documentation is not for you but instead for the data consumers. Figure out language, visualizations, and descriptions that work for teams across the organization so end users are able to properly find what they are looking for and easily use it.

Rolling Out Data Documentation

When first rolling out your data documentation, you should experiment with what works by doing smaller tests and documentation and getting feedback. Here are a few other useful tips to help you along the way:

User experience

If you can, pair with user experience or product experts for help. Run interviews with your users and determine if the documentation is useful. Iterate and improve with this feedback.

Standardize only after success

Standardize and roll out full data documentation only when you have a system that is working and one that you can maintain and improve over time.

Maintain accuracy
: Inaccurate documentation is often worse than no documentation at all, because your users will think they understand the data, but they don't. If they end up building models or making decisions based on this misunderstanding, it can end in disaster. This means you need to build something sustainable that is as easy to maintain as possible.

Just like with code and architecture documentation, your organization and data users will reap the rewards of well-documented data. This will expedite projects, standardize how the organization works with data, and help clarify data lineage and quality for easier decision-making. It will also bring up privacy and security questions at the beginning of the data lifecycle and at the start of new projects, where they are most effective!

You may already have a good grip on documentation but one nagging problem, the presence of undocumented data, that is unknown to you. It's important to tackle this problem, as undocumented data is often sensitive in nature.

Finding and Documenting Unknown Data

Unknown data are gaps in data documentation or even basic knowledge and understanding of the data. These occur at large organizations due to lack of best practices, usually over the course of many years. Data from past applications or sunsetted products, data amassed via acquisitions and partnerships, or data purchased long ago accumulate in undocumented databases or files. Sometimes this data is in active use, but no one knows how it got there or when it was collected. Other times the data is newly discovered, and the company is unclear on its origin. These types of undocumented data can also be the result of knowledge loss—where data was not documented before key people left.

When dealing with unknown data, it's important to establish a process and routine so as to prevent the data from sitting around even longer without documentation. Unknown, unmanaged, and untracked data presents large privacy and security risks, which you'll learn later in this chapter as well as in Chapter 4. Here is a suggested process you can use to investigate, document, and determine a decision when you find unknown data. Feel free to modify it to fit your needs by adding steps or clearer requirements for each stage, such as the specific location and technology that should be used.

1. *Investigate potential provenance.*
   Does the data look like any already documented data at the company? Is the data easily found by searching a search engine (i.e., publicly available data)? Does anyone on a related team know where it came from?

2. *Dive into discovery.*
   Investigate all potential sources and connect with other teams and departments. Someone might know the origin of the data or might see similarities with data they have used. Look at the contents of the data for potential clues and document your findings as you go.

3. *Determine sensitivity.*
   Does the data have person-related or personally identifiable information? Can you determine the date it was collected by looking at any related dates in the database or document? If person-related, what would your company's privacy policy say regarding the data you found? Is there a particular data privacy or sensitivity classification that should be noted and maintained?

4. *Start documentation for consumption.*
   How can you build documentation people will read and use? To start, document where the data was found, what it includes, and what you were able to find when asking about provenance and the sensitivity levels. Documentation should follow organizational standards and should be discoverable and accessible. At this point, you may want to involve data management decision makers to determine next steps.

5. *Delete, archive, or maintain?*
   What are the next steps for the "now known" data? To decide, involve interested parties, including compliance and audit departments if they exist at your organization. If the data isn't sensitive, doesn't relate to any proprietary details or people, and is useful, you can likely just integrate it. Ensure the documentation is shared and updated and move on. Otherwise, you might choose to archive the data until more details become available. This is also a good option if the data isn't useful, but you would like to wait a given time before deleting it. Data minimization standards for data privacy recommend deleting the data, particularly if it contains personal details and you were not able to ascertain under what circumstances it was acquired. You should still document this decision and the investigation before you delete it so there is an audit trail. I would almost always suggest deletion if the data appears too old to be valuable for data science or business purposes.

There are several products that help teams find unknown data, if you are particularly concerned about its existence at your company. These services often provide scanning software that looks at servers and attempts to find data where you expect none. That said, if you are properly documenting your data processes and are working closely with other data teams, it's unlikely that data will just sit around unused and undiscovered.

One common issue with unknown data is that it is historical reporting data, often collected and used by business decisions makers before data science was established

at the company. If the data science team or focus at your company is fairly new, then it might be worth investigating reporting data used by business units that falls outside your normal data collection mechanisms. For example, there could be many spreadsheets or other types of document-based reports that were used for years before there was better data available on customers or products. There could also be integrations with tools like employee or customer management systems or other software that pull data and place it onto internal servers or filesystems. Familiarize yourself with these if you join a new team concerned about undocumented data sources.

Sometimes these practices have developed due to *shadow IT*, where sensitive data is copied to many locations because of access restrictions. Shadow IT is a term used to describe processes outside of IT leadership or purview—often created as helpful shortcuts—that frequently lead to security and auditing nightmares. Sadly, this process is commonplace, because humans spend lots of hours, days, and weeks waiting for access to be granted. Once access is granted, users immediately copy the data or develop automation to do so, so they don't have to wait again. Part of your job as a privacy-focused data person will be to identify these practices and build better privacy technologies to expedite future access. Replace shadow IT with transparent, easy-to-use, well-documented, and privacy-preserving access systems!

You may encounter a variety of reactions to your search for unknown data as well as your recommendations to delete it. Some folks might be defensive, afraid, or resistant to the approach; however, it's important for the business to ensure data rights are reliably and verifiably respected—regardless of good intentions. Some data teams have an idea that data should be saved, no matter what, and this type of data hoarding presents significant privacy challenges.

Rather than tackle the cultural and communication problem alone, educate decision makers about the risks of harboring unknown data. Spreadsheets full of customer or employee information are valuable assets, which either should be documented and managed by capable data and security teams following best practices or should be deleted so as to not become a target for internal or external security threats. If these reports are useful for answering questions, move the data into the light and keep it documented and audited. You will also get better insights and high-quality decisions by doing so!

Undocumented data is often orphaned data in systems with no or little lineage tracking. Similar to the basic documentation you've learned thus far, data lineage is an essential tool for data governance.

Tracking Data Lineage

Data lineage (sometimes called *data provenance*) is a way of tracking where the data came from, how it got there, and what actions have been performed on the data since entering the system. As you can imagine, this information is extremely useful for data scientists to clarify data quality, data content, and data utility.

Lineage information helps you answer questions like these:

- When was this data collected? From where?
- How am I allowed to use this data? What consent was given at collection?
- What processing, cleaning, and preparation has this data undergone (i.e., removal of nulls, standardization of units, etc.)?
- Where else is this data processed, and where is related data stored?
- What should I keep in mind about the quality and origin of the data?

Unfortunately, when companies originally developed their data infrastructure, adequate systems to track data lineage were often not available. The focus tended to be on ingesting the data and storing it away as efficiently as possible—not on tracing it and determining how it was processed. Technological debt builds up around data systems. If you find there is no lineage data, you can always start now.

Depending on how advanced your data infrastructure and engineering systems are, there might already be great places to begin documenting lineage. This information can be pulled from systems like Apache Spark (or Beam, Flink, Kafka, Airflow) or other pipeline automation systems and integrated into whatever documentation or tracking systems you are using. If you aren't sure what's already being done, it's a good idea to connect with the teams who manage data catalogs and data schema. A data catalog is an index with documentation on data sources managed and made available across the company, often including data documentation, access requirements, and even processing, storage, and quality information. If your company doesn't currently catalog data or track lineage information, then you should coordinate a group of data folks and determine the most lightweight approach to getting started.

As a data scientist, you've probably investigated a dataset that makes you question its validity. Only via proper lineage or provenance tracking can you ascertain if errors were made in collection or processing. Knowing when and where the error occurred can help you and other team members determine what might have caused it and whether there's a bug involved. Tracking origin and processing is essential when maintaining streaming or near real-time systems, as errors in the data collection and transformation can quickly propagate into models and other downstream products.

There are now many tools around data documentation, lineage, and version control. Ideally, you can find a good set of tools or even a single tool to help manage the governance concerns you have. There are always newcomers, but some tools I've seen teams enjoy are DBT, CKAN, AWS Glue, and Tableau (with their Data Catalog).

To monitor changes in data flows, I recommend using tools like Great Expectations, which can be used to test data midstream and determine if anything has markedly changed. Great Expectations allows you to write what I call *data unit tests*, to assert that the data meets your expectations. For example, you may want to test if a particular value is not null, that a value is above or below a certain number, that a date string has been properly standardized, or even that a value is a string or an integer. These tests, when used properly by data teams, can immediately alert you of aforementioned bugs.

There are also clear wins for privacy when implementing lineage data. You'll learn more about consent tracking in Chapter 3, but a lineage and consent tracking pipeline could look like Figure 1-2. First, the user is presented with an interface with fine-grained privacy preferences explained in clear language. As the data is collected, provenance details like where, when, and how the data was collected are inserted into the data structure at the same level as the data as database fields, instead of being attached in a separate JSON document that no one uses.[5] The data enters the normal cleaning and transformation pipelines. That data is then analyzed for quality and other governance standards. Data sensitivity or the presence of PII is also analyzed in a semi-automated manner. This step may need to occur as the initial step depending on the data source and the logging included in the transformation steps. Please make sure to not log sensitive data! The data will then be stored, and the additional information is saved in linkable data structures. This process ensures additional governance data is easily referable and remains current and attached to the user data until the data is removed from the system.

5 Ideally this is either a separate easily linkable table or the rows themselves have extra columns or attributes, making lookup easy. You will see a concrete example of this in Chapter 3.

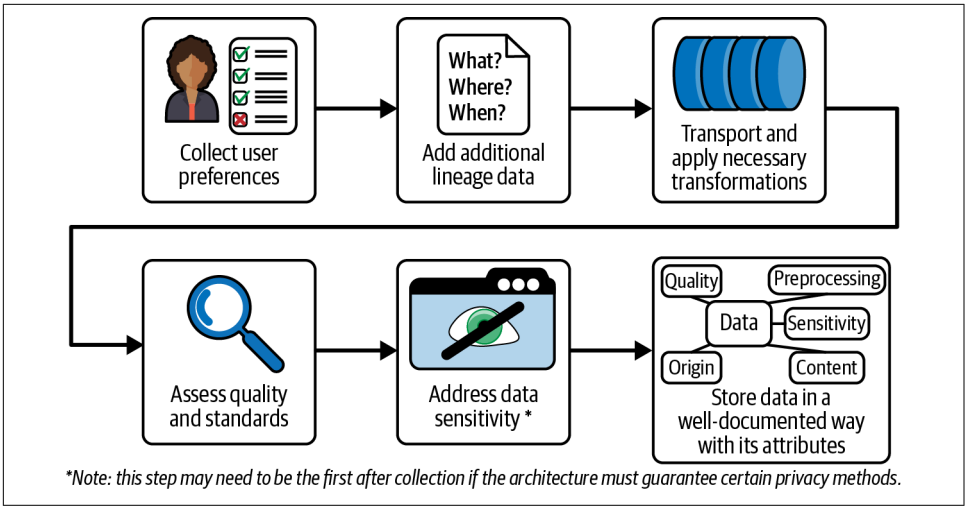


Figure 1-2. Lineage and consent tracking pipeline

Lineage also addresses data sovereignty legislation and policies, which seem to be increasing over time. These laws focus on keeping residents' data inside a particular jurisdiction, such as keeping EU resident data hosted on infrastructure in the EU. Ensuring sovereignty controls are in place can assist your legal, compliance, and security teams in validating that the data infrastructure is following the law.

Tracking data lineage and consent adds overhead now but helps debugging and speeds up access to data later. If you work closely with a data engineering or infrastructure team, talk with them about sharing the workload for setting up and maintaining lineage information. Just like your data documentation, ensure it's readable, interpretable, and usable for the data consumers. If it sits in a file that no one ever looks at, it's not worth producing and maintaining in the first place!

Data Version Control

When you work with data daily, you understand incremental data changes occur. If you have a very large dataset, these changes are not noticeable. These incremental changes have a larger effect and impact workflows, analytics, and modeling when the data collection and transformations significantly shift or when external forces impact the incoming data, like during a global pandemic.

This is when data versioning comes into play. Data versioning is the ability to create versions or checkpoints for your data tied to a particular time. Similar to code version control systems like `git`, data version control allows you to monitor changes by creating a snapshot or "commit" at a particular point in time. This can then be compared with earlier or later versions to understand how the data changed.

Similar to agile software development practices such as testing, continuous integration (CI), continuous delivery (CD), and software version control, data version control benefits data science. Imagine being able to pinpoint when a change or bug happened in order to diagnose why a model isn't performing as expected or why a particular report or analytics tool stopped working.

Knowing how data you collect changes over time can be a superpower for understanding the behavior or systems you are modeling. How are the users of your application changing over time? How are your experiment results shifting? What assumptions did you have about the data originally, and how have they held up? Many questions you ask about the data can benefit from a periodic review to understand the data itself and its changes over time. Changes in software, pipelines, or other processing can significantly shift and introduce errors into the data. This is important to monitor with the potential to roll back the data and software as it might negatively affect other data models, analysis, and systems.

Data versioning also helps data privacy and trustworthy AI practices. When you can determine what data was used for a particular model, when you can pin a before and after point when someone requested their data be removed, and when you can transparently demonstrate the changes you have made to preserve privacy, you can ensure and audit responsible data use in the system. Data versioning supports each of these actions and helps ensure that you are clear about the outcomes of the data privacy measures. Then you can debug them should anything go wrong.

So how do you start versioning data? There are several tools available. Here are some evaluation questions to find what versioning tools are appropriate for you:

- How does the tool manage snapshots and checkpoints for the data? This should be done in a way that is both programmable and well understood. You don't want to be reading through documentation when you need to recover data immediately!
- Are the snapshots or versions memory efficient? How will you manage space? A fairly naive way to manage data would be to copy your entire database every day and save an old copy. This is great if you have unlimited memory and space and only a few hundred rows, but that's fairly abnormal. You'll want to evaluate how many snapshots you save, how much extra space and processing power those will take, and when to delete old snapshots.
- Will this integrate well with other teams and their workflows? As with all software choices, you should ensure that what works for your team also helps other teams working with the data. Check with the data engineers and software engineers to ensure they also know how to "recover" data. You should ensure that they understand both the API and programming language and how the tool is used.

- Can you imagine how this tool would be used? Have your team play with the tools and write up a few example stories to see if you can pre-program relevant use cases. How would a data recovery after a schema change work? How could you answer what data was used for a particular model that was trained and deployed? What if a GDPR deletion request comes in and you want to prove the data was properly removed? Spend time with the data team to make a comprehensive list and ensure the use cases you want to use it for are well understood and maybe even programmed!
- Can this be used in conjunction with data lineage efforts to better select data for particular use cases? As you learned earlier in this chapter, understanding lineage can help determine if a particular dataset is well suited for the task at hand. This information, combined with version control, can speed up the type of modeling and experimentation by increasing the data understanding and finding shifts or changes best as soon as possible.

Data versioning is closely related to model versioning. The prior questions can also be asked in relation to model versioning tools, and several open source libraries today do both. Whatever your plan is and however your team works best, think about incorporating versioning for data and models into normal workflows. These practices are well understood in software and can help make the data science work in your organization more predictable, error-free, and better understood. Setting them up now and having them evolve with data governance usage is something I would recommend for every team, even if the first few years are a learning process.

There is also growing support for version control of datasets directly in data lake and data warehousing tools. If your organization is already using large data management tools, it'd be a good idea to first check if there is some support or integration before introducing another library.[6]

As you are likely aware, the data tooling landscape changes quickly. Having a look around at what new tools are available and evaluating several before deciding what is best is always a good idea! Use these questions to guide your evaluation and selection criteria and don't be afraid to do a few proof-of-concept implementations before setting up one as the standard.

6 If the organization is on an older on-premise setup without these features, you might think about a low-tech solution like regular snapshots of data used for particular tasks and some tools to easily load or exchange snapshots as required. There is also expanded support for self-hosted and on-premise setups from many of the versioning tools, including DVC.

Basic Privacy: Pseudonymization for Privacy by Design

You've learned about what data governance is, how to find and evaluate sensitive data, how to document data with regard to its sensitivity, and how to figure out when things change via governance tracking and version control. With this command of data governance, you can now begin to apply privacy techniques for person-related data in well-documented and repeatable ways.

To begin, start with the basics. Sometimes the simplest approach can solve your problems and address numerous internal and external concerns. You'll be evaluating which privacy technologies and techniques are appropriate for the risks at hand and what are the rewards (for example, granting expanded access to the data).

Pseudonymization is a great match for basic privacy needs, like when you are dealing with a data use case where the data will never be exposed to someone outside of a trusted group of employees. Pseudonymization is a technique that allows you to use "pseudonyms" instead of real names and data. There are several approaches to pseudonymization, outlined in Table 1-2.

Table 1-2. Approaches to pseudonymization

Pseudonymization approach	Description	Example
Masking	Applying a "mask" to the data that often replaces values with a standard series of values.	888-23-5322 → <ID-NUMBER> or <XXX-XX-5322>
Tokenization (table-based)	Replacing identifiable tokens via a lookup table that allows a one-to-one replacement.	Mondo Bamber → Fiona Molyn
Hashing	Using a hash mechanism to make the data less interpretable but still linkable.	*foo@bar.com* → 32dz22945nzow
Format-preserving encryption	Using a cipher or other cryptographic technique to replace the data with similar data. Often this is also linkable.	(0)30 4344 3333 → (0)44 4627 1111

As you may already notice in Table 1-2, these approaches can significantly vary the quality of your data as well as the privacy of the individuals. For example, the hashing mechanism takes what is easily interpreted as an email address and turns it into something that is no longer interpretable. This provides minimal privacy but also destroys our ability to extract useful information (such as linking email accounts based on domain). Depending on the implementation, masking can either remove all identifying information or leave too much information where it is easily linked with other datasets to reveal personal information. Table-based tokenization means maintaining a solution that may not scale with your data but that allows appropriate and human-readable linking if you need to connect disparate datasets.

As you'll explore further in Chapter 4, linking is a primary attack vector to determine the identity of an individual. The more data you are able to link, the easier it is to use

that linked information to infer who the person is or to learn enough about them to make a good guess. Format-preserving encryption retains linkability but is more scalable due to using a standard two-way mechanism based in cryptographic techniques. The linkability can often be destroyed at arranged time intervals by changing the secret key material. This can provide enough security for internal use cases as long as reasonable defaults are used. If linkability is what you are after, you should also consider varied techniques in the field of Privacy-Preserving Record Linkage (PPRL), which covers these pseudonymization methods along with several probabilistic hash methods.

Table 1-3 summarizes the benefits and drawbacks of pseudonymization.

Table 1-3. Benefits and drawbacks of pseudonymization

Benefit	Drawback
Linkable: Pseudonymization techniques often retain the ability to link data, which helps when you are connecting datasets using personal identifiers or other sensitive columns.	Pseudonymization is not anonymization. Re-identification of pseudonymized data via linkage attacks are a prominent and consistent threat to privacy protection and become easier the more data is available (more on these in Chapter 4).
Format-preserving utility: Several pseudonymization methods allow you to preserve format or learn the original intention for that data (i.e., is it an email?). This can be helpful if you have questions about the original data source and makeup.	Any information included in the pseudonymized data is already more information and more risk should the data get released or accidentally exposed. Would data documentation be a better way to understand the underlying schema?
Privacy by Design technique: Pseudonymization is a technique proposed by frameworks like Privacy by Design to provide useful alternatives to using the raw data, especially when this data is sensitive.	Basic techniques like pseudonymization can create a false sense of security, making people more likely to share the data more widely or to claim it is "anonymized" because personal identifiers are removed or pseudonymized.

In my experience, the biggest argument against using pseudonymization would be that it creates a false sense of security and privacy. I have seen teams struggle with privacy solutions and determine that pseudonymization is good enough because it "anonymizes" the data. This is unfortunately a widespread myth. You'll learn what anonymization is and isn't in Chapter 2, but I can guarantee you that no amount of pseudonymization will bring you the anonymization you seek, should that be the recommended method.

However, if you can guarantee the data will be used only internally by a small group of individuals who may require privileged access, then pseudonymization might be a good fit. One use case might be for internal sales or customer support teams, who need to see customer details but likely don't need access to all of the information. Another could be for internal business intelligence (BI) and analytics dashboards, which need to link data but should not have direct access to the sensitive values.

In both of these cases, however, there are clear alternatives to pseudonymization. You could imagine the BI Dashboard reporting orders across geographies would only

need aggregated queries for large-enough geographies to ensure some privacy is preserved. Or you can imagine a customer support system that doesn't expose fields that are not necessary to perform the task at hand.

I've often seen pseudonymization used as a mechanism to extract data from production systems and use it in testing environments (for software development and analytics tools) or to log potential errors in a secure system. This has the benefit that the data is "close" to the data seen in production but is not the actual raw production data. As you can imagine, having this production data in a testing environment is extremely risky and should not happen, since the data is only weakly protected and test environments are often insecure. If production-data properties are required for a test to pass (for example, for model testing), I recommend that you find a way to synthetically generate those properties in your testing data rather than use real production data with minimal privacy protections.

If pseudonymization is recommended by your internal stakeholders and you have also deemed the risk low enough for its use, there are several tools and libraries to evaluate. I also have a notebook in the book's repository where I walk through some silly and fun examples of pseudonymization.

Here, I'll present an example workflow using Hashicorp Vault. Hashicorp Vault is a service used by infrastructure teams to manage secrets across applications. This is a common pattern for microservice setups, where many applications and services are deployed in containers and need to access sensitive data such as API keys, encryption keys, or identity details in a scalable and secure way.

To use the format-preserving mechanism, you would first build a regular expression pattern for the format you need. Here is an example of a regular expression for a credit card number:

```
\d{4}-\d{2}(\d{2})-(\d{4})-(\d{4})
```

You register this pattern as a template for a particular transformation in Hashicorp with a set of roles attached (i.e., who is allowed to use this transformation). Hashicorp already supports certain types of format-preserving encryption. Note: some of these methods are reversible, and some are not!

You can test your transformation by using the command-line interface (CLI) and seeing that a fake generated number comes back as a different but still valid credit card number:

```
$ vault write transform/encode/payments value=1111-2222-3333-4444

Key              Value
---              -----
encoded_value    1111-2200-1452-4879
```

If you chose a method that is reversible, you can also check that you can properly decode with the proper role and permissions:

```
$ vault write transform/decode/payments value=1111-2200-1452-4879

Key             Value
---             -----
decoded_value   1111-2222-3333-4444
```

The API calls may change and need to be cleared with the infrastructure support running Hashicorp for your organization. I recommend taking a look at the latest Hashicorp documentation to see if they are the right solution for your pseudonymization needs.

There are also several open source libraries with format-preserving encryption support as well as other pseudonymization methods (such as hashing, masking, or tokenization). Here are a few with useful documentation and functionality at the writing of this book:

- KIProtect's Kodex provides an open source community edition that has several pseudonymization methods.[7]
- Format-Preserving Encryption Python library by Mysto allows you to set up several format-preserving algorithms with an easy-to-use Python interface.
- Microsoft's Presidio allows for several masking and tokenization options as well as measures to discover PII in text.
- Private Input Masked Output (PIMO) uses a Go-based engine and has many templates to pseudonymize and mask data.

As you explore the advanced techniques and potential threats in this book, you'll develop a better understanding of what risk level and ease of use fits your team and work. In doing so, it will become clear to you when pseudonymization fits and when it's best to approach the problem with a more advanced and protective technique.

Summary

This chapter outlined what data governance is and how you can and should use it as a data scientist. You learned important avenues that link governance to data science and data privacy, such as finding undocumented data, identifying sensitive data,

7 Disclaimer: I cofounded KI Protect and worked on the initial implementations of this library; however, the company now runs without my involvement, and I am no longer a contributor to the library.

managing data documentation, and tracking lineage. You also learned some basic privacy approaches for working with sensitive data, such as pseudonymization.

You should feel like you are getting your foot in the door. You are building your mental map of privacy and what is important and relevant for your work. You may already have some questions about how to practically identify and manage privacy risk or how to ensure you are using privacy techniques effectively. I have some good news: the next chapter helps you think about these questions using scientific methods. Let's move on to differential privacy!

About the Author

Katharine Jarmul is a privacy activist and data scientist whose work and research focuses on privacy and security in data science workflows. She has held numerous leadership and independent contributor roles at large companies and startups in the US and Germany—implementing data processing and machine learning systems with privacy and security built in and developing forward-looking, privacy-first data strategy. She is a passionate and internationally recognized data scientist, programmer, and lecturer.

Colophon

The animal on the cover of *Practical Data Privacy* is a species of morid cod called *Eretmophorus kleinenbergi*. Not much is known about this fish, and some scientists believe it may not be a unique animal at all but rather the juvenile stage of another morid species. It is found in the Mediterranean Sea as well as parts of the Atlantic Ocean, and grows to be about 3.5 inches long.

Morid cods, of the family *Moridae*, are also known as codlings, hakelings, or moras. They are small- to medium-sized fishes, generally with large eyes, a chin barbell, two dorsal fins (one short and triangular, the other long), and an elongated body tapering to a narrow tail. Their diet is made up of a variety of plankton, invertebrates, smaller fish, and crustaceans. They have been found at depths up to 8,200 feet but tend to live in shallower water.

Morid cods are classified in the same taxonomic order as cods. While they share some external similarities, morids differ in skeletal structure and the shape of their swim bladder (an internal organ filled with gas that allows fish to adjust their buoyancy and remain at their current depth without the effort of swimming).

Many of the animals on O'Reilly covers are endangered; all of them are important to the world.

The cover illustration is by Karen Montgomery, based on an antique line engraving from a loose plate, origin unknown. The cover fonts are Gilroy Semibold and Guardian Sans. The text font is Adobe Minion Pro; the heading font is Adobe Myriad Condensed; and the code font is Dalton Maag's Ubuntu Mono.