O'REILLY®

Automating Data Quality Monitoring at Scale

Going Deeper than Data Observability

Early Release
Raw & Unedited

Compliments of
Anomalo

Jeremy Stanley &
Paige Schwartz

Anomalo

LEARN MORE

Automatically detect data issues and understand their root causes, before anyone else.

Are data quality issues taking a toll on your team?
Anomalo helps you get ahead of data issues by automatically detecting them as soon as they appear in your data and before anyone else is impacted.

DETECT

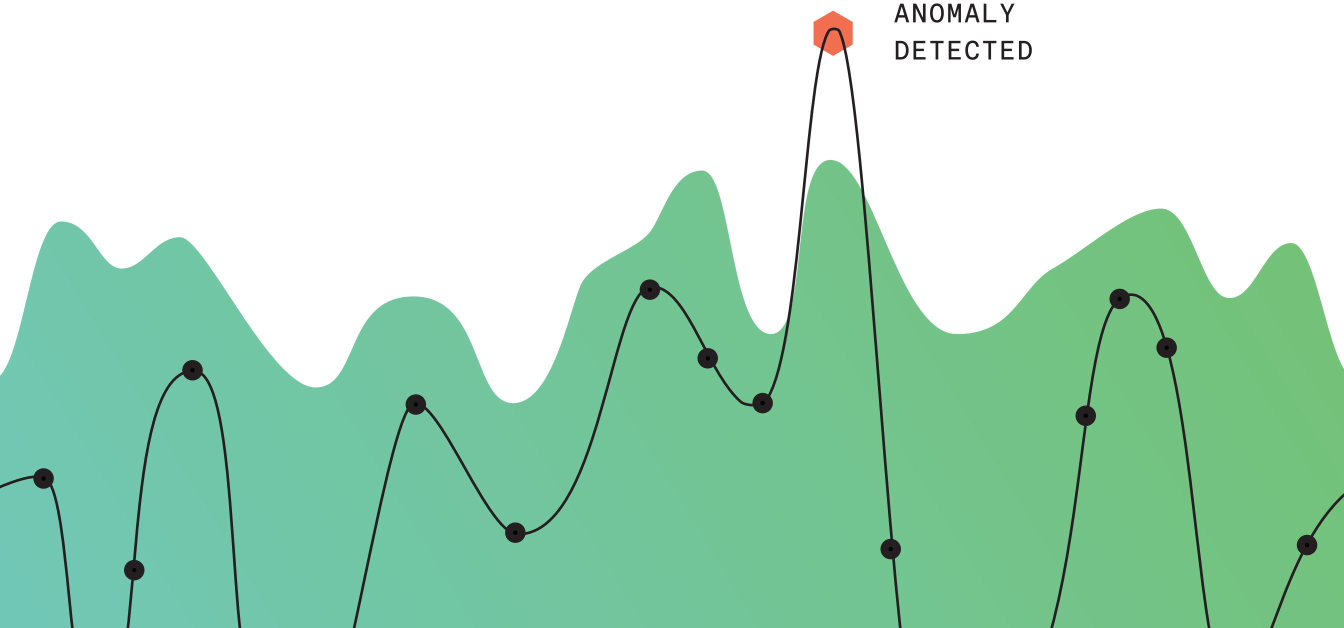
Go beyond metadata observability, with deep data quality you can trust.

ALERT

The right alerts, sent to the right person. No noise.

RESOLVE

Save hours of time investigating an issue with root cause analysis.



Automating Data Quality Monitoring at Scale

Going Deeper than Data Observability

With Early Release ebooks, you get books in their earliest form—the authors' raw and unedited content as they write—so you can take advantage of these technologies long before the official release of these titles.

Jeremy Stanley and Paige Schwartz

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

Automating Data Quality Monitoring at Scale

by Jeremy Stanley and Paige Schwartz

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: Aaron Black
Development Editor: Gary O'Brien
Production Editor: Elizabeth Faerm
Copyeditor: TO COME
Proofreader: TO COME

Indexer: TO COME
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Kate Dullea

December 2023: First Edition

Revision History for the Early Release
2023-02-21: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781098145934* for release details.

The views expressed in this work are those of the authors and do not represent the publisher's views. While the publisher and the authors have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the authors disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

This work is part of a collaboration between O'Reilly and Anomalo. See our statement of editorial independence.

978-1-098-14587-3

[LSI]

Table of Contents

CHAPTER 1
The Data Quality Imperative

A Note for Early Release Readers

With Early Release ebooks, you get books in their earliest form—the authors' raw and unedited content as they write—so you can take advantage of these technologies long before the official release of these titles.

This will be the 1st chapter of the final book.

If you have comments about how we might improve the content and/or examples in this book, or if you notice missing material within this chapter, please reach out to the editor at *gobrien@oreilly.com*.

In March 2022, Equifax was migrating its data from on-premise systems to a new cloud infrastructure, a notoriously tricky process—complex legacy data pipelines need to be replicated in a new environment. Somewhere along the way, an error was introduced, impacting how credit scores were calculated. Roughly 12% of all the company's credit score data was affected, and hundreds of thousands of people ended up with scores that were off by 25 points or more. Unaware of the error, lending institutions altered the rates they offered customers and even rejected loan and mortgage applications that should have been approved.

Unfortunately, this isn't the only data quality mess that's made the news recently.

- In 2020, a data error caused the loss of 16,000 positive COVID test results in the UK, possibly resulting in 50,000 people not being told to self-isolate.
- So-called airline "mistake fares," discounted by more than 90%, cost airline companies money and reputation if they fail to honor these "glitch prices."

- Facebook provided a data set to a group of social scientists that left out half of all its U.S. users, affecting the findings in academic work studying the impact of social media on elections and democracy.
- The video game company Unity lost $110 million on their AI advertising system after ingesting bad data training from a third party.

These news stories show the impact that particularly bad data quality issues can have—but that's not the whole picture. Most data quality issues happen on a smaller scale but build up, eroding trust in the data over time. Worse, the vast majority of data quality issues are never caught and are sneakily destroying value for companies as you read this. Of those that are caught, few are publicly disclosed.

Here are two anecdotes that data teams might find all too relatable. The first is from a large technology company. One day, one of their product dashboards indicated a sudden drop in NPS (Net Promotor Score) survey results. This most likely meant that customers were frustrated about a change in the product, so it set off alarm bells up and down the organization. Senior leadership became involved; a task force was created. Engineers and analysts evaluated all the code changes from the past month and combed through all their user data to figure out what could have caused the plummeting scores.

The root cause was finally uncovered by a data scientist who was analyzing the NPS score data. In their analysis, they discovered that any scores of a 9 or 10 (out of 10) had stopped appearing entirely in the latest results. This then led to an investigation by engineering, who discovered that the NPS survey was embedded in an iframe that had *cropped off the highest NPS response values*—such that it was physically (or should we say digitally?) impossible for a customer to answer 9 or 10 on the survey.

Data quality issues don't just affect dashboards; increasingly, they're affecting machine learning models that impact end users and production systems. The following story from a ride-sharing company is a prime example. They had built an ML model to detect potentially fraudulent new rider accounts and automatically block them from signing up. As a key input to their model, they used third-party credit card data to predict the likelihood of chargebacks—when a rider makes a charge that's later disputed, costing the company money. The model had learned was that when the data from the third party was NULL, the likelihood of chargebacks was higher; it was a sign that there wasn't much information about the card's legitimacy.

Everything was working well until one day, the third party had a data quality issue that caused it to send NULL data much more often than it had before. No one noticed the error and the company continued using the ML model to make fraud predictions. This led to many new users being denied their ability to sign up for the company's services, as they were incorrectly classified as fraudulent accounts.

We bet every reader who works with data has had similar experiences. When data's done right, it unlocks incredible value. But when you have no quality assurance about your data, it's like trying to run a restaurant with ingredients that may or may not be contaminated—you might get lucky and no one will get sick, but some of the time, your customers and your business are liable to suffer. (And you can bet they won't look at your food the same way again.) One study found that 91% of IT decision-makers think they need to improve data quality at their company. 77% said they lack trust in their organization's business data.

Is there a way to bring back trust in data? And can you ensure that the kinds of issues we've just mentioned are detected immediately and resolved quickly, before anyone else is impacted, even (and especially) when you work with large volumes of complex data? We believe the answer is yes, and the solution is automated data quality monitoring. This book will help you discover the power of automated data quality monitoring and learn how to implement the most advanced systems and tools that will help ensure high-quality, trustworthy data for your business.

High-Quality Data is More Important for Businesses Than Ever

To understand the state of data quality, and why new and better solutions are needed, it's important to take stock of the changes in how businesses work with data. It feels like only yesterday that businesses were struggling to get data out of isolated databases. Then Hadoop made it possible to run more advanced queries on large and complex datasets—if you knew what you were doing. Now, cloud data warehouses/lakehouses and transformation tools like dbt are making it easier for data teams to store, query, manipulate, and share data.

These changes have expanded both the scope of the data itself and how the data is used. Data was once limited to a small set of operational, financial reporting, and risk information—a "walled garden" of critical resources that was tightly controlled and had limited access. Now, every scrap of data an organization touches is logged and loaded into the cloud data warehouse. And any decision maker in the organization can either (a) have access to interactive dashboards and reports to answer questions with data or (b) directly query the data themselves to make decisions. Furthermore, machine learning algorithms are increasingly used to automate or inform decisions across the enterprise, and to build new products like personalized recommendations or fraud detection.

We are living in what Silicon Valley investment firm Andreesen Horowitz has called the "decade of data." To see why, let's look at the following trends.

Industries disrupted by data-driven companies

The common storyline is that the fastest-growing, most successful companies today are software companies. But we'd argue that they're actually data companies. Consider Amazon, which built the world's largest retail platform not just by having world-class software engineering talent, but by figuring out how to harness data for personalized recommendations, real-time pricing, and optimized logistics. Or, to pick another industry, Capital One is one of the first U.S. banks to move its data from on-premise systems to the cloud. They've been able to differentiate by using data to personalize marketing to accelerate growth and make intelligent underwriting decisions to mitigate risk.

Today, the intersection of data and software is the competitive frontier in nearly every industry, whether you look at financial services (banking, insurance), commerce (retail, online marketplaces), digital media (social, publishing, platforms), or healthcare (consumer devices, medical records).

The democratization of data analytics

Eager to keep up with these disruptors—or to become one—companies are investing in using more data across more dimensions of the business. Analytics experts are increasingly embedded into functional units (marketing, growth, finance, product teams, etc.) to drive more sophisticated uses of data for decision-making. Their job is to ask questions like: Can we pull data on our customers' past browsing and purchase activity so that we can better tailor an email to them? Or: Can we look at how our power users are adopting the latest feature to see if our launch was successful?

There are now a plethora of tools that let analysts—or truly anyone on a cross-functional team—self-serve the answers to data-related questions without writing any code. In seconds, they can spin up a dashboard or report that would have taken an engineer a month to build not so long ago. To support these analytics needs, data is no longer maintained by a small centralized team or available as a consolidated fact table for the entire business. Instead, data is dispersed and managed by a wider group of people that sit closer to the business lines. These are exciting developments, but when more people interact with the data, it opens up more ways for data quality to go wrong (which we'll address in the next chapter).

The rise of AI/machine learning

Many companies have AI/ML on their roadmap because it can create incredible value in the form of personalized and automated interactions. Machine learning (or ML, which we will use interchangeably with AI) relies on advanced statistical models to predict the future based on historical signals in the data called *features*. With enough feature data and the right modeling techniques, AI can optimize or personalize any frequent interaction with an entity the business cares about (consumers, content, transactions, etc.).

Data quality makes or breaks ML models. You must ensure you have high-quality datasets for both training and inference. Models are quite good at performing well if the data they see in production matches the distribution of data they were trained upon. But models will tend to fail miserably when presented with data that is far outside of the distribution they have seen before. (Contrast this to humans, who can use higher-order intelligence to generalize from one domain or distribution to another and account for significant departures from the norm.)

Modern data management tools

No summary of data trends today would be complete without a mention of the modern data stack (even though we can't wait for this term to go out of style, to be completely honest!). Today, the right set of SaaS vendors can accomplish what a 100-person team of full-time data engineers would have done 10 years ago. Businesses can leverage more data, more easily, than before, but this only benefits the business if the data itself is high-quality.

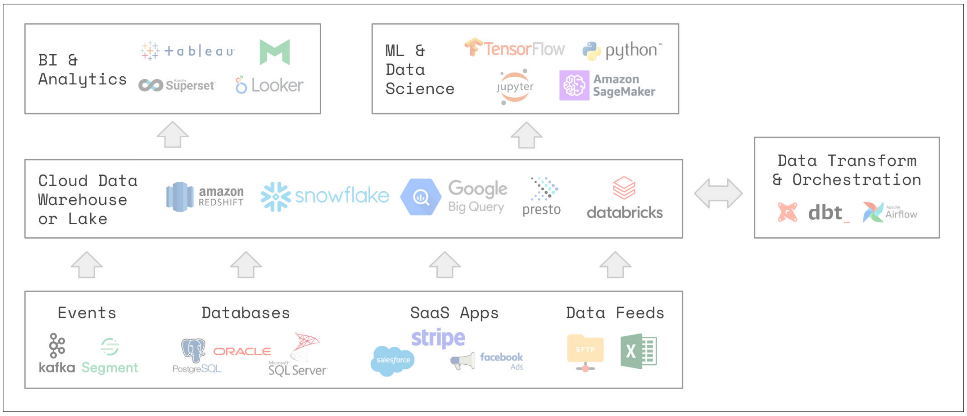


Figure 1-1. The modern data stack

Data Factories, Not Data Warehouses

Today, most companies have data factories, not data warehouses. We aren't loading indistinguishable boxes of data into storage units, to be neatly stacked and sent out for delivery. Instead, we feed raw data onto complex assembly lines, where it's transformed, customized, and manufactured into ever-evolving data products that go out to internal and external consumers.

Under the data mesh paradigm, which increasingly reflects how teams work with data at scale, producers are responsible for defining their products' boundaries, APIs, and data contracts. Data consumers and producers have different concerns and share the responsibility for data quality. Consumers will often have the most at stake, and are best able to triage issues. Producers have the most context on the factors that affect data quality, and typically are the ones who must fix any issues.

The Exponential Growth of Data Quality Risks

In the previous section, we explained the trends driving businesses to depend on high-quality data more than ever. The unfortunate truth is that high-quality data is harder than ever to achieve. The reason isn't that companies have gotten lazier or engineers don't care about data quality. We'd argue that *almost all data begins as high quality*. When a product is first built and instrumented, the data captured about that product by the engineer who built it is usually very closely aligned with their intention and the product's function.

But data does not exist in a vacuum. Over time, complexity creeps into the product and the data associated with it, as the product interacts with other systems and is used in new and different ways. The bottom line is that today there is more complexity happening faster and with fewer guardrails. Plus, performing a large-scale migration or using third-party data sources, both facts of life for most businesses today, can further increase your "data entropy"—and downgrade your data quality. In this section, we'll examine some of the biggest factors driving data quality degradation.

Outages or changes in the data factory

As companies' data factories become more complicated, they're more susceptible to miscommunications, broken parts, or human error. Here are just a few examples of what can go wrong:

- A database runs out of memory and stops accepting writes or allowing data loads or transformations to execute
- The data loading platform has an outage, causing late or incomplete data
- A subset of data is loaded twice, leading to duplicate records
- An Airflow job is scheduled to run too early or too late, leading to missing or corrupted data
- A SQL case/when statement in a transformation is changed such that it is impossible for records to satisfy some of the conditions, leading to an invalid distribution of categories

Isssues inside the data factory are often the most common sources of data quality incidents, as they directly affect the flow and contents of the data (and can be very difficult to test outside of a production data environment).

Migration from on-prem to cloud

The transition from on-premises data warehouses to cloud/SaaS providers is a common source of data quality issues.

Many companies have very complex data pipelines originating from mainframe legacy systems that were transferred into increasingly legacy on-prem data warehouses (Teradata, Hadoop, etc.).These legacy systems have been in place for decades, and have accumulated a tremendous amount of incremental complexity as new features have been added, migrations have been made, teams have come and gone, and/or business requirements have changed.

When moving this tangled web of data processing and storage into the cloud, companies seek to replicate what was done in their on-prem environment. But there can be very subtle issues introduced in recreating the on-prem flows in the cloud, leading to major data quality consequences.

For example, one company we work with mentioned that their customers' dates of birth were mangled badly. In a legacy mainframe, birthdates were stored as integers offset from a certain reference date, like Jan 1, 1900. Upon export, those integers were then converted into dates in the new cloud warehouse, but using the UNIX timestamp reference of 1970 as the offset. So all the birthdates were pushed far into the future.

The company had a marketing application that would send emails to customers based on their age—and it ended up sending no emails at all once it was pointed to the cloud version of the customer data pipeline. (Marketing teams often end up bearing a significant amount of pain from poor data quality. One survey found that marketers were wasting 21% of their total budget on data issues.)

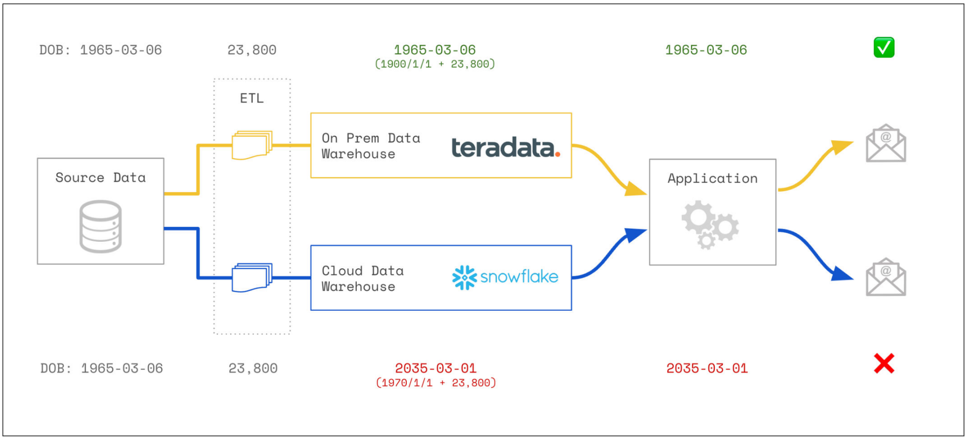


Figure 1-2. When data stored in an on-prem data warehouse is migrated to a cloud data warehouse, discrepancies can arise, such as this example involving different reference dates for calculating customers' birthdates.

Proliferation of third-party data sources

Using third-party data—that is, data that comes from outside the company—is easier and more common than ever. To give some examples, while I (Jeremy) was at Instacart, third-party data we regularly collected included:

- Weather data for demand forecasting/scheduling
- 3rd party services for maps information for routing
- CPG product catalog data for enriching the user search and shopping experience
- Fraud propensity scoring data for avoiding chargebacks
- Retailer inventory data for determining what's on the shelf at each store location at a given time

Frequently, third-party data is codified as a data relationship between two partners: Company A and Company B have to work together to service customer X or achieve operation Y, and that requires exchanging data. Sometimes, this is called second-party data.

In other cases, you are consuming public data or data packaged by a third-party service provider to be resold in a one-to-many relationship, often to make decisions about entities (customers, companies, locations) about which you have limited information. If you browse the publicly available feeds on the online catalog Demyst, you'll see that it's possible to leverage comprehensive tax data, property data, and business data all with just a few clicks.

Third-party data is a common source of issues. Not only could the provider make a mistake, but when the data contracts between provider and consumer are unclear or nonexistent, a misunderstanding could occur—for instance, you could be leveraging the data in a way that they didn't anticipate, causing them to make a change that accidentally breaks your use case. Similarly, SaaS tools you use for source-of-truth data could change the API or data format at any time.

Company growth/change

We noted earlier that data begins as high-quality. And it can remain that way, but only if the data is hermetically sealed where no one can touch it—which isn't how data at most companies works. In the real world, a company is constantly adapting and improving its products, which in turn affects the data emitted by those products. These are the primary reasons that data quality changes:

New features

New features often expand the scope of what data the system is logging. Insofar as this is an "add additional columns" type of change, the data quality risk isn't high. However, in some cases, new features may replace existing functionality, so the data emitted by the system can suddenly change. In many cases, the new features may change the shape of how data is being logged. The granularity level might be increased—such as logging at the level of an item rather than the level of the entire product. Or, what was previously logged in a single message may be broken apart and restructured into many messages.

Bug fixes

The average piece of commercial software contains 20–30 bugs for every 1,000 lines of code, according to Carnegie Mellon University's CyLab Sustainable Computing Consortium. Bug fixes can have the same impact as new features. They can also genuinely improve data quality—but when that sudden improvement comes as a "shock" to the systems that depend upon the data, there may be negative consequences. (More on shocks later in this chapter.)

Refactors

Refactoring happens when teams want to improve the structure of the code or systems behind an application without changing the functionality. However, refactors often present the risk of unintended changes—especially to things like data capture which may not be robustly tested in the application code.

Optimizations

Frequently, changes are made simply to improve the speed or efficiency of the performance of an application. In many cases, how data is being logged can be a performance issue, and changes can affect the reliability, temporal granularity, or uniqueness of the data emitted by the system.

New teams

New teams often inherit a legacy application and arrive with a limited understanding of how it interacts with other systems or how the data it produces is consumed. This increases the risk of all of the above.

Outages

In addition to intentional changes, many systems will simply have outages where they stop functioning or function in a degraded level of service. Data capture is often lost entirely during these outages. By itself, this is often not a data quality issue per se, as the lack of data is a reflection of the lack of activity due to the outage. But in many cases, the outage may affect the data being emitted without affecting the service itself, which *is* a data quality issue.

A Tangled Web of Data

In practice, we don't consider systems in isolation when we think about data. Instead, each product is part of a web of many systems that interact with each other by passing data back and forth (in real-time or by batching up requests). Therefore, what a system emits as data is a function not only of its behavior but also of *how it interacts with all the other systems in the network it is connected to*.

Engineers and product managers making a change typically understand the implications of that change on their own systems. But what about the implications that spread throughout the application or organization? Teams' abilities to understand and foresee these implications will decrease as we move further away from their area of ownership.

For example, if the product catalog team at an e-commerce company made a change to how their data is structured, they might understand how this would affect the company's product recommendation systems, which they work with frequently. But how likely are they to have insight into the implications for the advertising team selling targeted ads? What about the consequences for the fulfillment team that's optimizing how orders are routed—and is part of a completely different division? We'll cover ownership and organizational challenges like this one, with suggested solutions, later in this book.

Exogenous factors

When you leverage data to make decisions or build products, there will always be factors that affect the data that are outside of your control, such as user behavior, global events, competitor actions, and supplier and market forces. Note that these aren't data quality issues per se—but they often look and feel like data quality issues, and may need to be handled in a similar way.

The Effect of Competitor Actions on Data

What are competitor actions? In some industries, companies react to competitive behavior by using data in real time. For example, e-commerce companies may be monitoring competitor price data, and in response, adjusting their prices almost instantaneously. Airlines do the same. This can mean that a sudden change in a competitor's behavior could cause a significant automated change in *your* behavior, and result in a "shock" to your data that might appear to be a data quality issue.

The COVID-19 pandemic offered a striking example of an exogenous factor having a sudden impact on data. Everyone had to treat the beginning months of COVID as a special case when analyzing user behavior. For example, Figure 1-3 shows how data about the number of miles logged for Chicago taxi trips changed dramatically in March 2020.

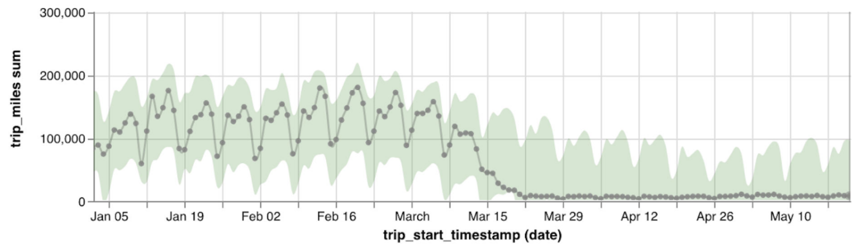


Figure 1-3. The average length of a taxi trip in Chicago plummeted in March 2020. The taxi data is publicly available from the city of Chicago.

Machine learning models had to be quickly retrained on fresh data, since their assumptions based on the historical trends no longer applied. In one well-known case, Zillow's model for predicting housing prices—which was the basis of a new business arm, Zillow Offers—couldn't adapt fast enough. The automated service overpaid for homes that it didn't end up being able to sell in the changing market, and sadly Zillow had to lay off almost a quarter of its workforce as a result.

Teams typically find themselves in one of two situations regarding external factors:

- In some cases, like with COVID-19, external changes are dramatic enough that you need to put your decision-making processes on notice and possibly retrain your ML models. It's almost like a data quality issue—it's a change that you want to be immediately notified about so that you can do damage control.

- In other cases, external factors have a more subtle influence, such as a supply-chain issue affecting your order processing times. You need to quickly understand the context of these changes and rule out any data quality problems, which can often look like real external trends.

Maintaining Data Quality is a Long-Term, Continuous Effort

Data quality issues are inevitable: they require a strategic approach that is proactive and constant rather than reactive and piecemeal. This is a marked difference from how most companies treat data quality today.

Why do most companies have a "whack-a-mole" approach to data quality? We'd argue it's because data is 10x more difficult to test than software. Code is the same today as it is tomorrow, barring a deliberate update. Data, on the other hand, is chaotic and constantly changing. You can test code in a controlled QA environment and also run unit tests that isolate just one part of the system. But you have to test data holistically in production, where it's subject to external factors you don't control, such as how users interact with your product in real time. Your tests have to be able to filter out these noisy factors and separate them from the true data quality signal.

For this reason, while software bugs are often quickly detected and fixed through automated testing and user feedback, *we strongly believe that the vast majority of data quality issues are never caught.* Because teams lack the right continuous monitoring tools for data, problems happen silently and go unnoticed.

Making matters worse, the cost of fixing a data quality issue increases dramatically the more time has passed since the issue occurred.

- The number of potential changes that could have caused the issue goes up linearly with the length of time over which you are evaluating.
- The amount of context the team has on why a change was made, or what the implications of that change could be, goes down with the time since the change.
- The cost to "fix" the issue (including back-filling the data) goes up with the amount of time since the issue was first introduced.
- Issues that persist for long periods of time end up becoming "normal behavior" to other downstream systems, so fixing them may cause new incidents.

When an incident is introduced and then fixed later, it really has two different types of impact. We call these data scars and data shocks.

Data scars and shocks

After an incident happens, unless the data is painstakingly repaired (which is often impossible or expensive to do), it will leave a *scar* in the data. We first heard this term used by Daniele Perito, Chief Data Officer & Co-Founder of Faire. A scar is a period of time for a given set of data where a subset of records are invalid or anomalous and cannot be trusted by any systems operating on those records in the future.

Data scars will impact ML models, as those models will have to adapt to learn different relationships in the data during the period of the scar. This will weaken their performance, and limit their ability to learn from all the data captured during the scar. It will also dampen the model's belief in the importance of the features affected by the scar—the model will underweight these inputs, wrongly believing they're less prevalent in the dataset. Even if you manage to go back in time and repair the scar, it's easy to introduce what's known as *data leakage* into downstream ML applications by inadvertently including some current state information in your fix. This leads to the model performing very well in offline evaluations (since it has access to "time-traveled" information from the future) but acting erratically in production (where it no longer has this information).

Data scars will also greatly impact any future analytics or data science work done on this dataset. They may lead to more complex data pipelines that are more difficult to write and maintain, as data users have to add a lot of exception handling to avoid biases introduced by the scar. These exceptions may need to be noted and addressed in any reporting or visualizations that include data from the time of the scar, increasing cognitive overhead on anyone trying to interpret the data or make decisions from it. Or, scars may need to be removed entirely from the dataset, leading to a "data amnesia" from that period, which can affect trend analysis or time-based comparisons (e.g., what was our year-over-year result for this statistic?).

In addition to the scarring effect, there are also effects in production that occur both when the data quality issue was introduced and when the data issue is fixed. This is what we call a data quality *shock*, and it can also affect AI/ML and decision-making.

When the data quality issue first occurs, any machine learning models that use features derived from the data will suddenly be presented with data that is entirely different from what they were trained on. This will cause them to be "shocked" by the new data, and they will produce predictions that are often wildly inaccurate for any observations affected by the data quality incident. This shock will last until the models are re-trained using new data, which often happens automatically in a continuous deployment model.

Then, once the data quality is fixed, that actually introduces yet another shock to the model (unless the data is repaired historically, which often isn't possible). The shock from the fix can often be as bad as the initial shock from the introduction of the data quality issue!

For analytics/reporting use cases, these shocks often manifest as metrics or analyses that have sudden unexpected changes. When these are observed, they are often mistaken for real-world changes (the whole purpose of these reports is to reflect what is happening in reality), so operations are changed or other decisions are made to respond to the data quality issue as though it was real. Again, the same thing can happen in reverse when the fix is released.

The longer the data quality issue goes unfixed, the deeper the scar, and the greater the shock from fixing it.

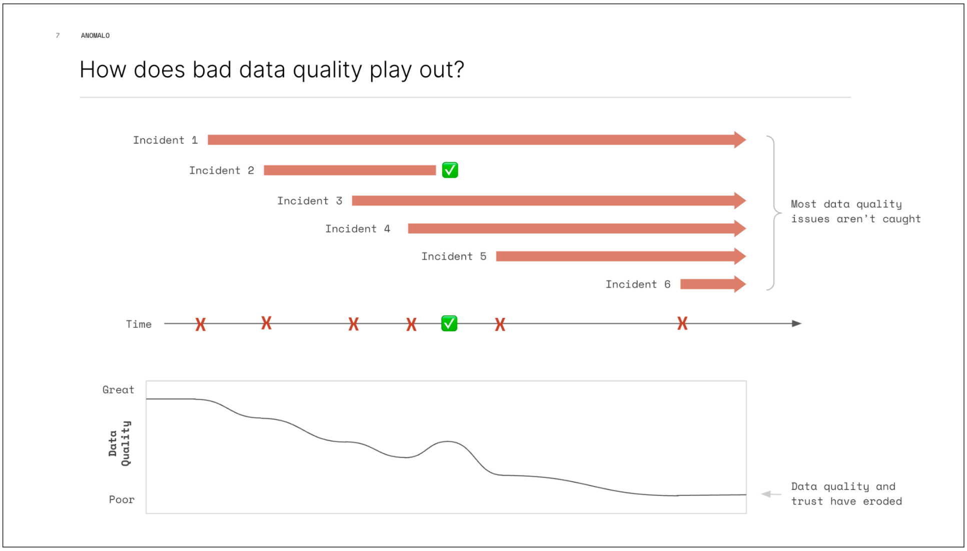


Figure 1-4. How incidents accumulate to erode data quality and trust over time. Each bar is a data scar left by the incident. Each X (marking when the incident first occurred) is a data shock. Notably, each checkmark (when the incident was resolved) is also a data shock.

The implication of allowing scars to continue accumulating is that slowly, over time, the objective quality of the data erodes. Figure 1-4 illustrates how incidents pile up such that, in the current state, everyone becomes convinced that the data quality is low and that the data itself is untrustworthy. And as hard as it is to backfill data, it's even harder to backfill trust.

Therefore, there's a framing change needed in how organizations think about combatting poor data quality. It shouldn't be a one-off project to go in and fix data quality for a given data source. It needs to instead be a continuous data quality monitoring initiative, where data quality issues are found as they occur and resolved as quickly as possible.

Sail Smoothly on the Sea of Data with Automated Data Quality Monitoring

Businesses today rely on data to ensure their product is both reliable and secure, and to out-innovate their competitors. If you've worked in data for more than a few years, you've witnessed these changes personally, as the modern data stack makes it possible to leverage a dizzying amount of internal and third-party data and quickly conduct powerful new kinds of analytics. Even more exciting, at a company with just a small data team you can now build sophisticated machine learning models for use cases from creating AIs for gaming to running real-time pricing engines.

Unfortunately, one tool is mostly missing from companies' new data stacks: an automated data quality monitoring solution that will protect against inevitable data errors and help them to be fixed quickly—before they cause scars in the data and shocks to production analytics and machine learning systems.

Lacking such a solution, businesses find themselves in an all-to-common state of affairs where data quality has eroded and overall trust in the data is low. This runs counter to the aims of businesses' digital transformation efforts, and all the money and time invested in making data easier to access and share. It's as if we've built a ship, planned our route, and departed on the open seas of data—but forgot to bring along a maintenance crew. Now our boat is leaking in multiple places, and we can't blame the crew for feeling a bit mutinous.

However, a lifeboat is here. Automated data quality monitoring can help you detect issues, triage bugs quickly, and address the root causes before anyone is affected. The upcoming chapters will show you how.

About the Authors

Jeremy Stanley is co-founder and CTO at Anomalo. Prior to Anomalo, Jeremy was the VP of Data Science at Instacart, where he led machine learning and drove multiple initiatives to improve the company's profitability. Previously, he led data science and engineering at other hyper-growth companies like Sailthru. He's applied machine learning and AI technologies to everything from insurance and accounting to ad-tech and last-mile delivery logistics. He's also a recognized thought leader in the data science community with hugely popular blog posts like Deep Learning with Emojis (not Math). Jeremy holds a BS in Mathematics from Wichita State University and an MBA from Columbia University.

Paige Schwartz is a professional technical writer at Anomalo who has worked with clients such as Airbnb, Grammarly, and Samsara, as well as successful startups like CodeSignal, Tecton, Clerky, and Fiddler. She specializes in communicating complex software engineering topics to a general audience and has spent her career working with machine learning and data systems, including 5 years as a product manager on Google Search. She holds a joint BA in Computer Science and English from UC Berkeley.