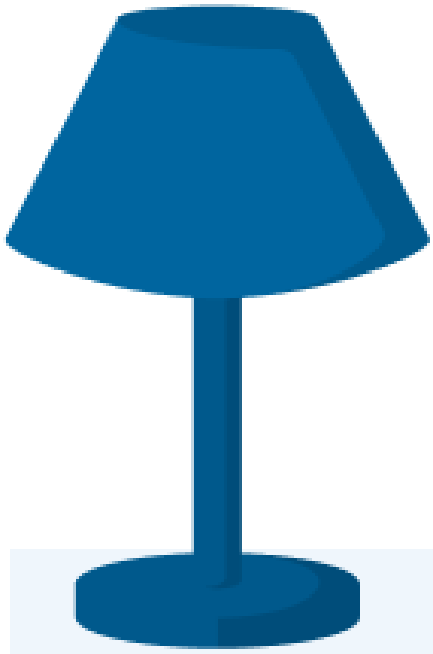
MONTE CARLO

Data Quality & Observability Evaluation Guide

A comprehensive guide to uncovering the best data reliability solutions for your organization.

Table of Contents



I. Introduction

The data estate has changed, providing data leaders with greater value - and greater complexity.

Now, enterprise teams are responsible for more:

- Diverse data sources
- Layers of data transformations
- Interdependent technologies
- Consumers
- Analytical, AI, or ML data products

Delivering trusted data in this context is difficult, especially with legacy data quality approaches. Similar to how software engineers turned to observability to deliver reliable applications, organizations must embrace data observability to deliver reliable data products.

Traditional data quality solutions evaluate data across the six dimensions – completeness, consistency, uniqueness, accuracy, validity, and timeliness – at a moment in time. But:

- The data upstream might be incorrect or late
- Issues may be rare, conditional, or persistent
- The root cause may be resolved quickly…or not

I. Introduction

Your team needs a modern approach that is designed to reduce the period of time data is not fit for use.

This approach is called data observability, and it's the only way organizations can resolve incidents at scale to achieve data trust and reduce risk.

Data observability combines data quality detection features with powerful resolution capabilities for full visibility into your entire data estate, including your data, systems, and code, and anomalous data is correlated to the root cause for lightning fast resolution. For example:

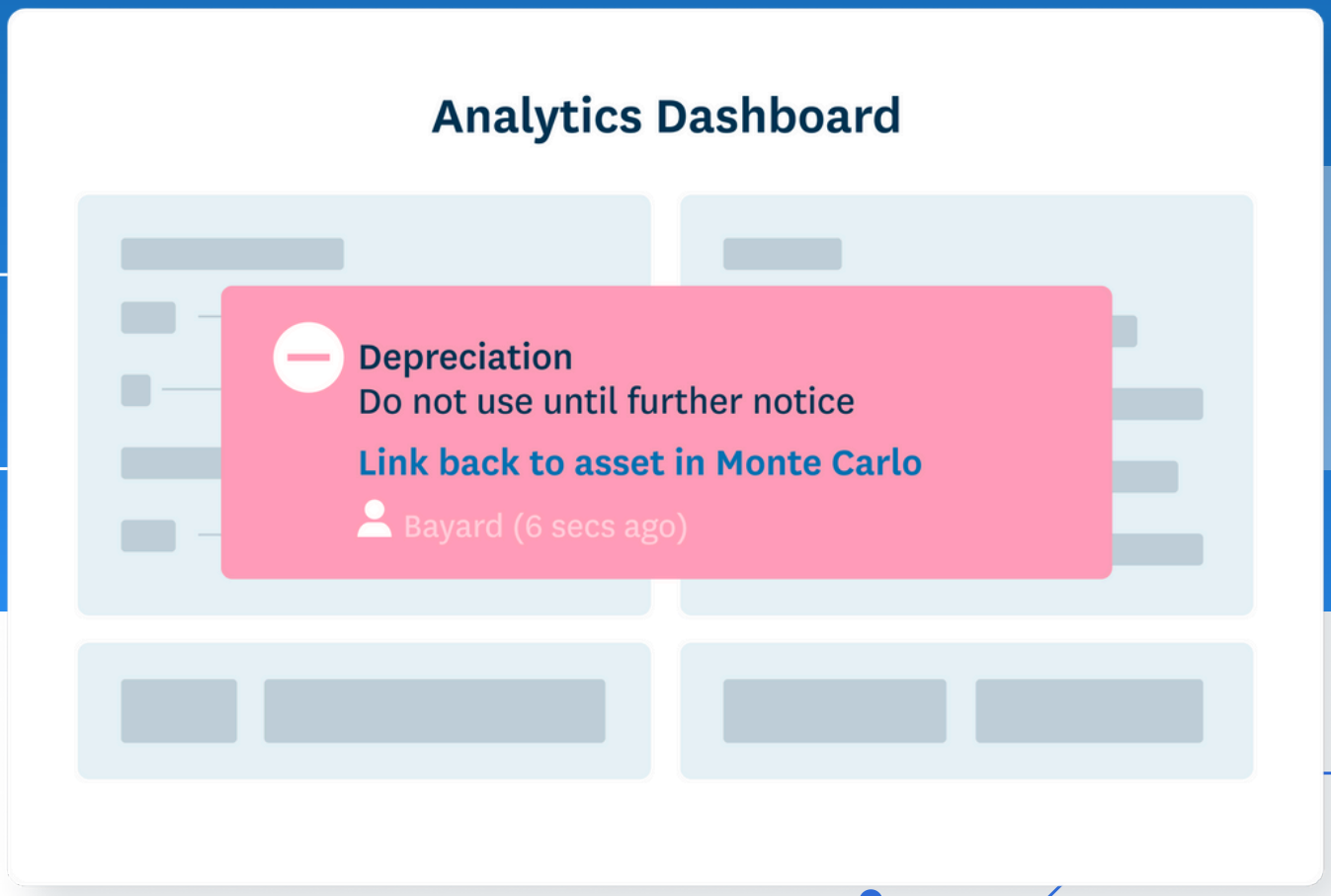
- DATA – what source sent us bad data?
- SYSTEM – what specific job failed?
- CODE – what code change led to the anomaly?

By providing AI-powered monitoring and incident management across your entire data estate, Monte Carlo enables you to take the next step on your data quality journey.

In this guide, we'll share the must-have reliability features you need based on analyst perspectives and first-hand experience—to demonstrate why a modern approach to data quality matters—plus an RFP template you can use to make choosing the right platform a little easier.

Let's get started!

II. The Key Components of Data Observability



Data Quality + Observability Evaluation Criteria

At it's core, a data observability solution takes data quality management to the next level by operationalizing detection, triage, and resolution of data issues through a mix of AI-powered and customizable data quality tools.

Comprehensive data observability tools will integrate across your entire data stack and provide coverage for issues like data freshness, volume anomalies, and schema changes from CI/CD in addition to monitoring your data directly on your most critical tables for things like NULL rates, duplicates, and values outside a normal distribution.

At Monte Carlo, our customers' needs are never far from our mind when we think about developing data observability as a category and our own feature roadmap. Considering the needs of our customers and the data industry at large, here are the key evaluation criteria we feel are the most critical to a comprehensive and future-proof data quality and observability solution:

- Enterprise readiness
- End-to-end coverage
- Seamless incident management
- Integrated data lineage
- Comprehensive root cause analysis
- Quick time-to-value
- AI observability

So, now that we've got those points in mind, let's dive into each of these criteria in a bit more detail.

Enterprise readiness

The world of data is always evolving. That's why you need a data observability provider that can serve as a strategic advisor. Any vendors can promise the world, but what can they actually deliver? Will a team of 12 people in a garage still be around in a year to observe your data? These are important questions to answer through customer reference calls to understand a solution's overall maturity.

Some key areas to evaluate for enterprise readiness include:

- Security– Do they have SOC II certification? Robust role based access controls?
- Architecture– Do they have multiple deployment options for the level of control over the connection? How does it impact data warehouse/lakehouse performance?
- Usability– Is an alert just pretty or will it actually save you time—like bulk update incidents or being able to deploy monitors-as-code. Usability can also be subjective in a POC, so it's important to balance this with perspective from actual users.
- Scalability– What are their largest deployments? Has this organization proven its ability to grow alongside its customers? Other key features here include things like ability to support domains, reporting, change logging, and more.
- Support– Data observability isn't just a technology, it's an operational process. The maturity of the vendor's customer success organization can impact your own level of success as can support SLAs (if the vendor doesn't have support SLAs, that's a red flag).
- Innovation history and roadmap– The data world changes rapidly and as we enter the AI era, you need a partner that has a history of first-to-market innovation. Fast followers are often anything but, with comparative features shipped 6 months to a year later. (That's 25 in chief data officer years.) Cloud-native solutions often have an advantage here.

End-to-end coverage

The true power of data observability lies in its ability to integrate across modern data platform layers to create end-to-end visibility into your critical pipelines.

For years, data testing–whether it was hardcoded, dbt tests, or some other type of unit test–was the primary mechanism to catch bad data.

While still relevant in the right context, the problem with data testing as a complete practice is that you couldn't possibly write a test for every single way your data could break. No matter how well you know your pipelines, unknown unknowns will still be a fact of life. And even if you could identify every potential break (which you can't), you certainly wouldn't be able to scale your testing to account for each one as your environment grew. That leaves a lot of cracks in your pipelines to fill.

Data quality and observability tools should offer both broad automated metadata monitoring across all the tables once they have been added to your selected schemas, as well as deep monitoring for issues inherent in the data itself.

A strong data observability tool will also integrate widely across your data platform, from ingestion to BI and consumption, and enable quick time-to-value through simple plug and play integrations.

Be sure to verify that your chosen solution offers tooling integrations for each of the layers you'll need to monitor in order to validate the quality of your data products, as well as integrations into existing workflows with tools like Slack, Microsoft Teams, Jira, and GitHub.

Incident management

Most data teams we talk to initially have a detection focused mind-set as it relates to data quality, likely formed from their experience with data testing.

The beauty of data observability is that not only can you catch more meaningful incidents, but the best solutions will also include features that improve and accelerate your ability to manage incidents. Bad data is inevitable and having tools to mitigate its impact provides tremendous value. There are a few areas to evaluate when it comes to incident management:

- Impact analysis — How do you know if an incident is critical and requires prioritizing? Easy—you look at the impact. Data observability tools that provide automated column-level lineage out-of-the-box will also sometimes provide an impact radius dashboard to illustrate how far a quality issue has extended from its root. This can help data engineers understand at a glance how many teams or products have been impacted by a particular issue and who needs to be kept informed as it moves through triage and resolution.

- Internal team collaboration — Once an alert has triggered there needs to be a process for assigning and potentially transferring ownership surrounding the incident. This may involve integrating with external ticket management solutions like JIRA or ServiceNow, or some teams may choose to manage the incident lifecycle within the data observability tool itself. Either way, it's helpful to have the flexibility to do both.

- Proactive communication with data consumers — When consumers use bad data to make decisions, you lose data trust. Data observability solutions should have means for proactively communicating with data consumers the current health of particular datasets or data products.

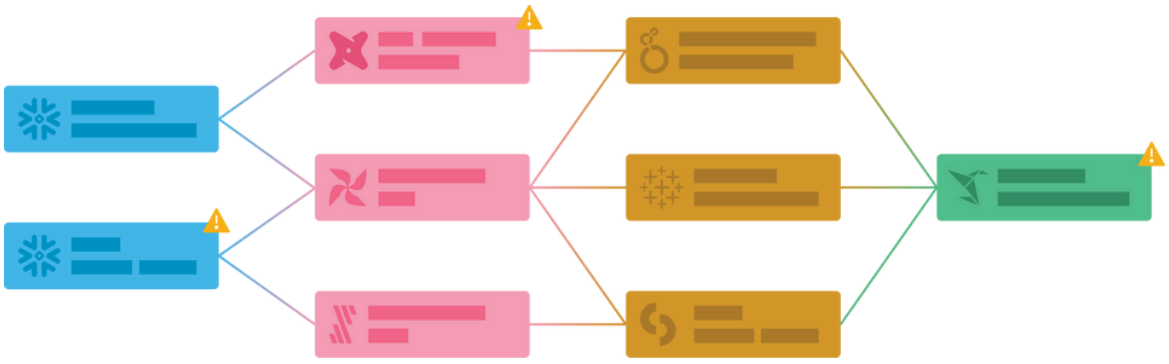
Integrated data lineage

Lineage is a dependency map that allows you to visualize the flow of data through your pipelines and simplify root cause analysis and remediation.

While a variety of tools like dbt will provide lineage mapping at the table level, very few extend that lineage into the columns of a table or show how that data flows across all of your systems. Sometimes called “field-level lineage,” column-level lineage maps the dependencies between data sets and tables across data products to understand visually how data moves through your pipelines.

It’s also important that your data lineage and data incident detection features work as an integrated solution within the same platform. A key reason for this is that lineage grouped alerting not only reduces alert fatigue, but helps tell a more cohesive story when an event impacts multiple tables.

Rather than getting 12 jumbled chapters that may be part of one or two stories, you are getting an alert with the full story and table of contents.



Root-cause analysis

What is your standard root cause analysis process? Does it feel disjointed hopping across multiple tools? How long does it take to resolve an issue?

Data can go bad in a lot of ways. A comprehensive data observability tool should help you identify if the root cause is an issue with the data, system, or code.

For example, the data can be bad from the source. If an application went buggy and you started seeing an abnormally low sales price from orders in New York, that would be considered a data issue.

Alternatively, a data environment is made up of a panoply of irreducibly complex systems that all need to work in tandem to deliver valuable data products for your downstream consumers. Sometimes the issue is hidden within this web of dependencies. If you had an Airflow job that caused your data to fail, the real culprit wouldn't be the data but a system issue.

Or if a bad dbt model or data warehouse query change ultimately broke the data product downstream, that would be considered a code issue.

A thorough data observability tool would be able to accurately identify these issues and provide the proper context to help your team remediate each at its source.

Quick Time To Value

Data observability is intended to reduce work—not to add more.

If a data observability tool is providing the right integrations and automated monitors for your environment out-of-the-box, it will be quick to implement and deliver near immediate time-to-value for data teams.

A data observability solution that requires more than an hour to get set up and more than a couple of days to start delivering value is unlikely to deliver the data quality efficiencies that a growing data organization would require to scale data quality long-term.

AI observability

Building differentiated, useful generative AI applications requires first party data. That means data engineers and high quality data are integral to the solution.

Most data observability solutions today will monitor the data pipelines powering RAG or fine tuning use cases–they are essentially the same as data pipelines powering other data products such as dashboards, ML applications, or customer facing data.

However, the generative AI ecosystem is evolving rapidly and your data observability vendor needs to be not just monitoring this evolution but helping to lead the charge. That means features like observability for vector databases, streaming data sources, and ensuring pipelines are as performant as possible.

III. Analyst Perspective

Gartner

While Gartner hasn't produced a data observability magic quadrant or report ranking data observability vendors, they have named it one of the hottest emerging technologies and placed it on the 2023 Data Management Hype Cycle.

They say data and analytics leaders should, "Explore the data observability tools available in the market by investigating their features, upfront setup, deployment models and possible constraints. Also consider how it fits to overall data ecosystems and how it interoperates with the existing tools."

We anticipate Gartner will continue to evolve and add to their guidance on data observability tools this year.

Ventana

The Ventana Research Buyers Guide does a good job capturing the essence of these tools saying, “data observability tools monitor not just the data in an individual environment for a specific purpose at a given point in time, but also the associated upstream and downstream data pipelines.”

They also used standard dimensions of SaaS platforms in how they ranked vendors:

- Adaptability
- Capability
- Manageability
- Reliability
- Usability
- Customer Experience
- TCO/ROI
- Validation

But, product capability is the highest weighted at 25% of the evaluation. Here, Ventana really hit the nail on the head saying that the best data observability solutions go beyond detection to focus on resolution, prevention and other workflows:

“The research largely focuses on how vendors apply data observability and the specific processes where some specialize, such as the detection of data reliability issues, compared to resolution and prevention. Vendors that have more breadth and depth and support the entire set of needs fared better than others. Vendors who specialize in the detection of data reliability issues did not perform as well as the others.”

G2 Crowd

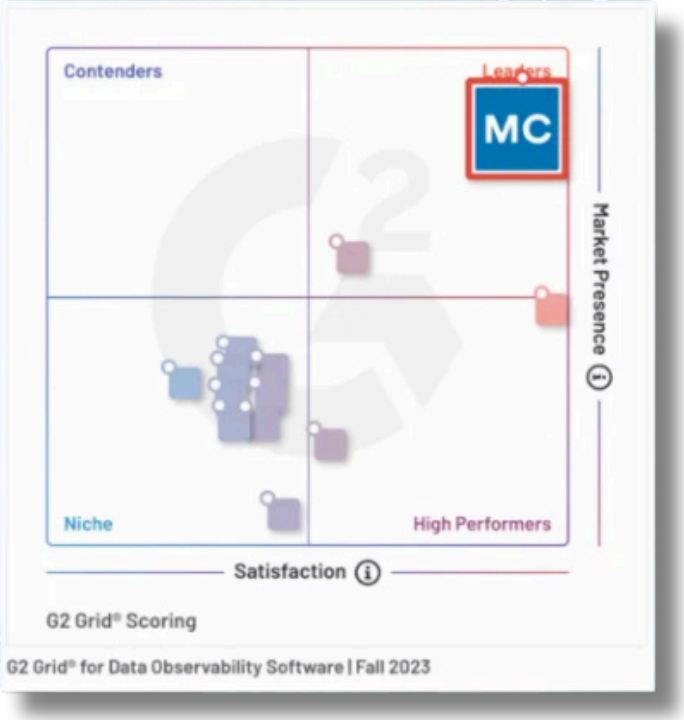
G2 was one of the earliest non-vendor resources to put together a credible list of data observability vendors and a definition for the category. They say:

To qualify for inclusion in the G2 Crowd data observability category, a product must:

- Proactively monitor, alert, track, log, compare, and analyze data for any errors or issues across the entire data stack
- Monitor data at rest and data in motion, and does not require data extraction from current storage location
- Connect to an existing stack without any need to write code or modify data pipelines

Vendors are evaluated by verified users of the product across a list of organizational and product specific capabilities including:

- Quality of support
- Ease of admin
- Ease of use
- Integrations
- Alerting
- Monitoring
- Product direction
- Automation
- Single pane view



IV. RFP Template

Data Quality + Observability Request For Proposals

Section	Key Capabilities	Criteria
Company Details	Vendor Experience	What is your experience in the industry? What references and case studies can you provide from similar projects?
	Innovation & Roadmap	What is your vision for the future of data observability? What are your planned enhancements and new features?
Security	Deployment Options	Is your product available in a software-as-a-service (SaaS) offering? Can you deploy the agent directly on our instance if desired?
	System Integration and Data Handling	Can you provide proof of SOC2 Type II certification? What options do you provide for authentication? Do you have role based access controls? Do you provide an API for retrieving security events to import into a SIEM? Are third-party tests available? What data is exported from our environment, and is that data encrypted? Is PII filtering available?
Configuration and Management	APIs	What functionality is available via API? Can monitors be created/configure via API?
	CLI & SDK	Is a command line interface available to simplify API interactions? Are any SDKs available, e.g. for use in Python scripts or Data Science Notebooks?
	Airflow Operator	Can monitoring and alerting be configured from within Airflow jobs without breaking a workflow?
	YAML	Can monitoring, alerting, and audiences be defined in YAML?
	Performance	How will your product impact our data warehouse/lake/lakehouse performance and compute costs?

RFP Continued		
Section	**Key Capabilities**	**Criteria**
Integrations	**Warehouse /Lake /Lakehouse**	What cloud native data warehouse and/or lakehouse technologies does your platform integrate with?
	Other Databases	What other database technologies does your platform integrate with?
	BI Tools	What BI technologies does your platform integrate with?
	Integration, Transformation, Orchestration	What ETL technologies does your platform integrate with?
	Collaboration Tools	What collaboration channels does your platform integrate with?
	Data Catalogs	What catalog technologies does your platform integrate with?
	Query Engines & Metastores	What query engines and metastores does your platform integrate with?
Support & CS	**Support & CS**	Do you provide web-based self-support resources? What is your support SLA? Are your releases backwards compatible? Do you charge additional fees for providing product support? What training, onboarding, and ongoing support is available?
	Pricing Structure	What is the basis of licenses for the product? Is on-demand/usage-based pricing available?
	Usability & Onboarding	How are deployment and product best practices shared amongst users? How is the time required to execute tasks minimized by the UI and key workflows? Is administration and management of the platform and its capabilities low-code or no-code?

Section	Key Capabilities	Criteria
ML Monitor & Anomaly Detection	**Available Monitors**	Does the product provide machine learning (ML) anomaly detection models: • to detect freshness anomalies? • to detect NULLs or missing values? • to determine volume anomalies, like table unchanged or when the number of rows added is too high or low based on historical patterns? • to alert when a column receives new values or an anomalous distribution of values in a column? • for % of unique values in a field? • for distribution, like min/max, average, stddev, variance, skew, percentiles? • for timestamp fields including detecting anomalies when the % of values in a field are in the future or past? • for validation, like detecting when the values in a column aren't in a standard format (email, social security number, US state code, etc.)? Does the product automate schema change detection (column name, data type)? Can ML monitors be built to monitor custom metrics with automated thresholds? Does the product allow users to monitor nested JSON in the field of a given table?
	Automation	What kinds of intelligent features are available to detect anomalies without the need for manual input? Can anomaly detection be automated to cover tables upon creation based on schema, database, domain, tag, and table name? Can anomaly detection be automated to cover specific pipelines, or data products, based on the selection of a downstream asset?
	Configuring Monitors	Are there mechanisms for adjusting the sensitivity of anomaly detection models? Can exclusion windows be set to ignore expected or seasonal data patterns? Can exclusion windows be automated for major holidays or seasonal events? Can ML monitors be adjusted to detect anomalies based on hourly, daily, or all record aggregations? Can ML monitors be scheduled? Can ML monitors be set to execute whenever a table is updated? Are configurations automatically recommended by the platform? Does the monitor creation process mandate a name and description?
	Deploying Monitors	Are specific models applied based on the specific table pattern type classification (I.E streaming_table / weekend_pattern / multimodal_update_pattern / etc.)? Approximately how many production tables are your ML monitors deployed across? Can ML monitors be deployed on specific table segments for one or more metrics? Can ML monitors be deployed without code for multiple variables (condition X AND condition Y = true)? Can volume anomaly detection be run on external views or tables? Can monitoring cost be allocated to specific query engines in the same warehouse? Are failure notifications sent when a monitor fails to execute successfully?

Section	Key Capabilities	Criteria
Data Validation Rules	Available Features	Does the platform offer pre-built data validations or rule templates where thresholds can be defined and deployed without code? Can data quality rules be deployed that compare values across tables? Does the platform offer the ability to identify records or key fields that are present in one dataset but missing in another? (referential integrity) Can data quality rules be deployed that compare values across database, warehouse, or lakehouse? (source to target checks) Does the platform offer the ability to profile a table interactively and without code?
	Configuring Rules	Can data validations and rules be executed based on a schedule or manual trigger? Can value-based thresholds be set to return a single numerical value when a specific column drops below a given value? Can custom SQL rules be generated with AI from within the platform? Does the platform offer the ability to automatically suggest data quality rules or ML monitors based on insights from data profiling and analysis?
	Deploying Monitors	Can data validations and rules be tested prior to deployment? Can custom data validations and rules with manually defined thresholds be created and deployed with SQL? Can complex data validations, with alert conditions featuring multiple variables, be created and deployed without code? Does the tool offer a circuit breaker which can stop pipelines when the data does not meet a set of quality or integrity thresholds?
Assets & Metadata	Metadata	What general information and metadata is listed within each asset? Are tags from other data systems, such as the data warehouse or ETL tools, surfaced and inherited within the platform? Is the schema, table type, and database name surfaced for each asset? Are usage statistics for each table including read/writes per day, # of users, latest updates, and # of dependencies automatically discovered and surfaced?
	Data Discovery	What types of data assets are discoverable within the platform? Are different table types—such as views, external tables, wildcard tables, dynamic tables—discoverable within the platform? Are non-table assets such as BI reports, orchestration jobs, and streams discoverable within the platform?
	Data Products	For each asset, does the platform surface: • monitor type and execution history? • number of reports and upstream/downstream dependencies? • historical update cadence, including time since last row count change? • row count history? • associated query logs? • performance of ETL systems acting on that specific asset?

Section	Key Capabilities	Criteria
Incident Management	Alerts	What alert channels are supported? Can alerts be muted? Can alerts for rule breaches that stay violated be adjusted to alert every time, after a certain number of runs, or only if the count of breached rows changes? Can alerts be routed to specific channels or audiences based on alert type, domain, asset importance, tag, table name, or dataset? Can alerts be sent in real-time or in a daily digest? Can alerts have pre-set priority levels? Can alerts be filtered by status, type, severity, tag, owner, table, schema, database, or audience? Can alerts have pre-set descriptions, including tags for specific owners or teams? What kind of guidance or initial information do system alerts provide? When validations/rules are breached, is the description of the validation, # of breached rows, and last breach date provided within the alert notification? Do alert notifications indicate the importance of an asset for triage? Are triggered alerts with the same dependencies (cascading incidents) automatically grouped within the same thread to reduce alert fatigue? Do alerts feature information on the specific downstream reports impacted (automatic impact analysis)? Are system alerts from ETL tools ingested and consolidated within the platform?
	Lineage	Is field level lineage surfaced for each asset? Is table level lineage surfaced for each asset? Does the platform's data lineage show ingestion, orchestration, and transformation workflows between lineage nodes?
	Incident Owners & Triage	Can the product automatically infer the priority of a detected issue by factors such as table popularity and consumption at the repository and BI layers? How is ownership of incidents and data assets tracked within the platform? Can alerts be escalated to incidents with different severity levels? Can alert/incident status be assigned and changed within the platform? How can data consumers be proactively alerted to incidents or the current health status of a dataset? What workflows and feeds exist for managing alert/incidents within the platform?
	Workflow & Visibility	What service/ticket management/workflow integrations exist for managing incidents outside of the platform? Is the history of data incidents or failed checks on an asset accessible? Can those incidents be annotated with custom notes? What data catalog integrations are available?

Section	Key Capabilities	Criteria
Resolution/ Root Cause Analysis	Insights	Does the platform: • detect insights to facilitate discovering the root cause of an incident? • provide automated suggestions for investigation queries based on how their team researched breaches of that rule in the past? • leverage metadata to surface the performance of orchestration jobs acting on specific assets for troubleshooting? • provide a unified incident timeline and systems performance dashboard to accelerate root cause analysis? • help prevent incidents by automatically surfacing the impact of a schema change on downstream assets during the pull request process?
	Data-level RCA	Does the platform provide segmentation analysis to automatically identify patterns within anomalous records? For example, if a spike in NULLS correlates to values of invoice_status=pending. Does the platform integrate multi-system data lineage with anomaly detection to help users pinpoint the origin of an incident? For ML monitors, does the product automatically provide anomalous record samples? For data validations, does the platform automatically provide breached row samples?
	System-level RCA	Does the product automatically surface system integration failures due to permissioning or credentialing issues? Does the product leverage metadata to automatically surface job failure alerts from ETL and orchestration systems? Does the product automatically correlate data anomalies to relevant system failures to accelerate root cause analysis?
	Code-level RCA	Does the platform surface the query logs of specific assets for troubleshooting? Does the platform automatically correlate anomalies to relevant changes in the query code of the underlying or upstream asset? Does the platform leverage metadata to correlate anomalies to failed queries? Does the platform leverage metadata to correlate anomalies to empty queries—queries that executed successfully but did not update or modify any data? Does the platform correlate data anomalies to pull requests (PRs) on relevant assets?
Reporting	Dashboards & Metrics	Does the product automatically: • compile alert and incident metrics (like status and severity) at the domain, dataset, audience, and data product levels? • compile operational response metrics (like time-to-response and time-to-fixed) at the domain, dataset, audience, and data product levels? • display data health (# of breached validations /custom rules) for specific tables? • create dashboards with all incident, data health, and operational response metrics across all tables within a selected dataset (data product pipeline)? Can metrics be easily exported via API?

V. The Future of Data Quality

What's next for data quality?

There's one critical feature that we didn't mention earlier, that plays a huge role in the long-term viability of a data observability solution.

And that's category leadership.

Like any piece of enterprise software, you aren't just making a decision for the here and now—you're making a bet on the future as well. When you choose a data observability solution, you're making a statement about the vision of that company and how closely it aligns to your own long-term goals. "Will this partner make the right decisions to continue to provide my organization with adequate data quality coverage in the future?"

Particularly as AI proliferates, having a solution that will innovate when and how you need it is equally as important as what that platform offers today.

Not only has Monte Carlo been named a proven category leader by the likes of G2, Gartner, Ventana, and the industry at large; but with a commitment to support vector databases for RAG and help organizations across industries power the future of market-ready enterprise AI, Monte Carlo has become the de facto leader for AI reliability as well.

There's no question that AI is a data product. And with a mission to power data quality for your most critical data products, Monte Carlo is committed to helping you deliver the most reliable and valuable AI products for your stakeholders.

Just starting your data observability journey?

Contact our team today to find out how Monte Carlo can help your team save time, reduce costs, and maximize your data resources with our category-creating Data Observability solution.

Check out more helpful resources on data and AI trends and best practices, including:

- Data Downtime Blog: Get fresh tips, how-tos, and expert advice on all things data.
- O'Reilly Data Quality Framework: The first several chapters of this practitioner's guide to building more trustworthy pipelines are free to access.
- Data Observability Product Tour: Check out this video tour showing just how a data observability platform works.
- Data Quality Value Calculator: Enter in a few specifics about your data environment and see how much you can save with data observability.