## 5 misconceptions about data observability and data reliability and why they are not true

Organizations that would benefit from implementing data observability solutions or data reliability engineering often do not do so due to some common misconceptions. In this blog post, we discuss five of them and provide clarity on each.

1. Data observability is only about monitoring

A common misconception about data observability is that it only involves monitoring data pipelines and systems. In reality, it also includes things like alerting, lineage, metadata, and metrics.

**Metrics:** Metrics are statistics calculated on data that are then monitored by data observability solutions. Some examples include:

* **%null** - percentage of rows in a column that are null
* **average** - mean of the value of the rows in a column
* **volume -** number of rows written in a table in the last 24 hours

Metrics are the core building block of every data observability solution. When a metric goes above or below certain thresholds considered “normal”, the data observability solution will classify it as a data quality issue.

**Anomaly Detection:** Anomaly detection refers to using historical data to understand dynamic patterns in data. For example, an anomaly detection algorithm might understand that a metric has a certain weekly seasonality, with peaks and troughs. A “normal” data point for Saturday night might be an abnormal one for Monday morning.

**Alerting:** It’s not sufficient to just monitor your data and detect problems. You also need to be notified when such a data issue arises. Data observability systems usually let you configure notifications either through email or Slack. To avoid being inundated with alerts, you need to make sure that your data observability solution’s anomaly detection is accurate, and more importantly, configurable.

**Lineage:** In data observability, lineage refers to the ability to trace data from its source to its destination through the data pipeline, identifying any changes or transformations that occur along the way. Lineage helps users to identify the cause of data issues and troubleshoot problems more effectively.

1. Data observability is a one-time implementation

Data observability isn’t just a one-time implementation – it’s a continuous process that requires maintenance, iteration, and learning.

For example, when onboarding to a new data observability solution, the data team will generally choose a set of data quality metrics to enable. However, this is not a set-it-once-and-done situation. It’s equally important to continue tracking and analyzing these metrics over time. As data systems change, some metrics may no longer be needed, while other ones become more important.

Another example is false positive alert feedback. When a data observability solution generates an alert, it’s crucial to investigate and determine whether it’s a true positive or a false positive (i.e. whether it indicates a real data issue). In the case of false positives, it’s important to provide feedback to the data observability solution to help it learn and avoid false positives in the future.

Fortunately, if you choose the right data observability solution, it should make it easy to provide this sort of maintenance: Bigeye, for example, automatically profiles any new tables in connected data sources, generating auto-metrics. In the case of false positives, Bigeye gives users the option to tell the algorithm to ignore that piece of data for anomaly detection training in the future.

1. Data observability is only for data scientists

While data scientists are one of the most prominent consumers of data in an organization, data observability isn’t just important for them. In reality, data observability is important for anyone who works with data or consumes data, including analysts, business users, and decision-makers.

1. You need to build your own data observability solution

Organizations often assume that they need to build a custom data observability solution in order to fit the nuances of their infrastructure. However, we generally recommend buying instead of building. Unless data quality is part of what your organization is selling, or you truly have complex, planet-scale bespoke infrastructure, purchasing off-the-shelf solutions is far more efficient. Especially if you are a company that is running on other pieces of the modern data stack - Airflow, DBT, Snowflake – off-the-shelf data observability tools should work very well for you.

The last few years have seen the proliferation of excellent data observability solutions on the market. These tools are often built by experts with decades of experience building large-scale data infrastructure projects, so they offer high levels of reliability and functionality, including sophisticated anomaly detection algorithms.

5. Using a data observability solution will be expensive and use up warehouse compute

One of the biggest concerns that organizations have about implementing a data observability solution is cost. As organizations have increasingly moved to ELT setups where data transformations are performed in-warehouse, every query run, whether for transformation or monitoring, incurs a cost. And while ELT warehouse costs are considered table stakes, many organizations balk at paying for data observability, considering it a “nice-to-have”. This is short-sighted thinking.

First off, the cost of implementing data observability is often far outweighed by the cost savings that result from early detection and remediation of data issues. If you monitor your raw tables as they land in the data warehouse, you are able to avoid running transformations over stale or incorrect data, and ultimately avoid expensive backfills and re-runs of pipelines.

Furthermore, data observability tools like Bigeye are optimized to run as few queries as possible:

* Bigeye only profiles all the tables once, upfront, to compute autometrics
* Freshness and volume metadata metrics are pulled from the warehouse logs, which do not incur costs
* Deeper columnar checks are up to the customer to enable
* Bigeye is rarely live-querying data. Instead, it caches the data from customers’ warehouses in its own databases.
* Bigeye performs clever batching of queries. For example, rather than running three separate queries for a max, a min, and an average, Bigeye will just run one query for all three.
* Bigeye runs queries at “low-traffic” times for data warehouses, and in a condensed fashion. Since Snowflake charges customers based on how long the warehouse is running, Bigeye tries to run all the Snowflake queries at the same time, rather than spread out across the entire day.

6. You can use the same tools for systems monitoring and observability as for data monitoring and observability

Since “data reliability engineering” and “data observability” are (intentionally) very similar terms to site reliability engineering and observability, executives at organizations may be tempted to think that you can just use the same tools for both. As an organization, if you’ve already invested in a solution like Datadog or ServiceNow for systems monitoring, why do you need something else?

While in both cases, the goal is to be able to fully observe the state of the system, the instrumentation and measurements of the state are different. For systems observability, you might want to know how many requests a service received and what percentage of those were errors. For data observability, though, that level of information might not be sufficient: a data pipeline might successfully complete without errors, but the data still has issues that need to be surfaced and alerted on. While you can probably make a tool like Datadog work for you in the data observability use case, it will require hack-ish engineering.

### Conclusion

Data reliability and data observability are easier and cheaper to implement than usually assumed - and more necessary. By understanding these misconceptions, organizations can make more informed decisions.