Observability at the long tail:

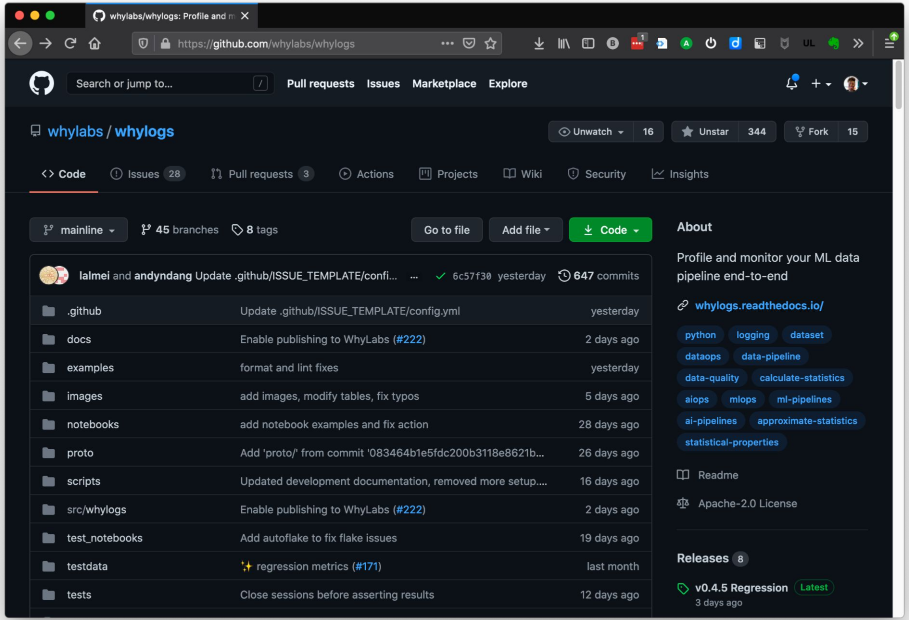
Why sampling production data doesn't work for rare events

Bernease Herman
Data Scientist, WhyLabs
Data Council Austin
March 23, 2022 in Austin, Texas

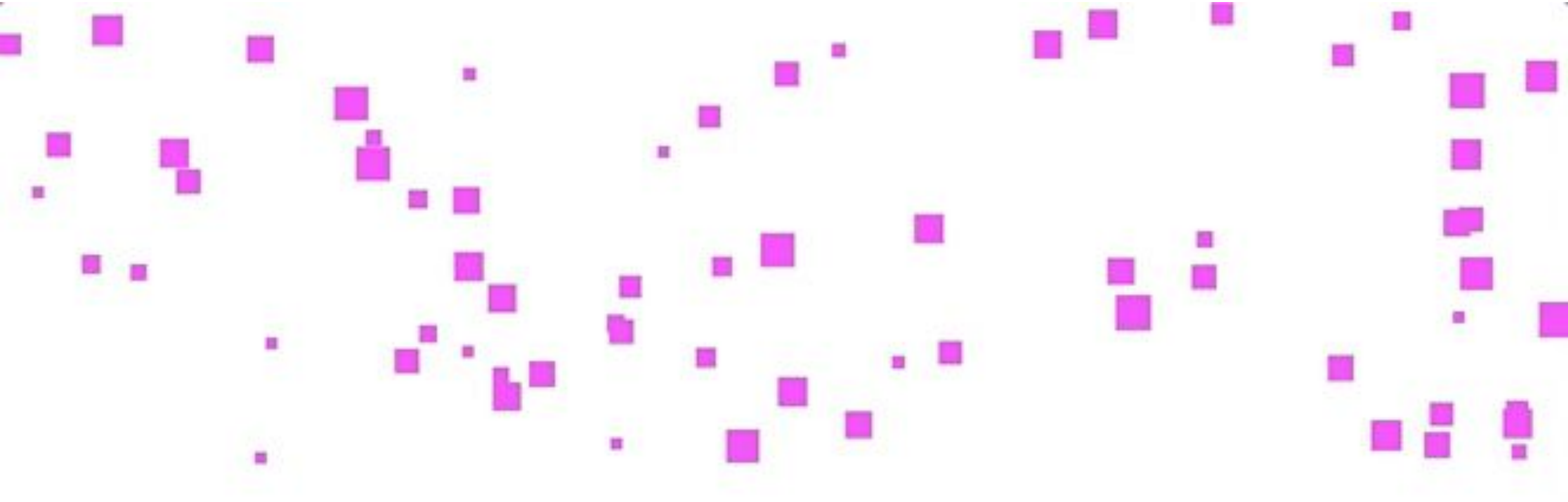
WHYLABS

On a mission to build the interface between human operators and AI applications bit.ly/whylogs:

Telemetry for the ML stack

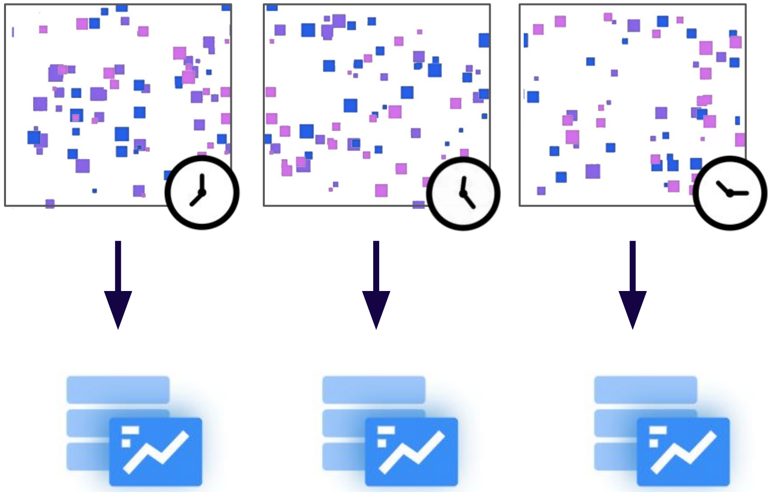


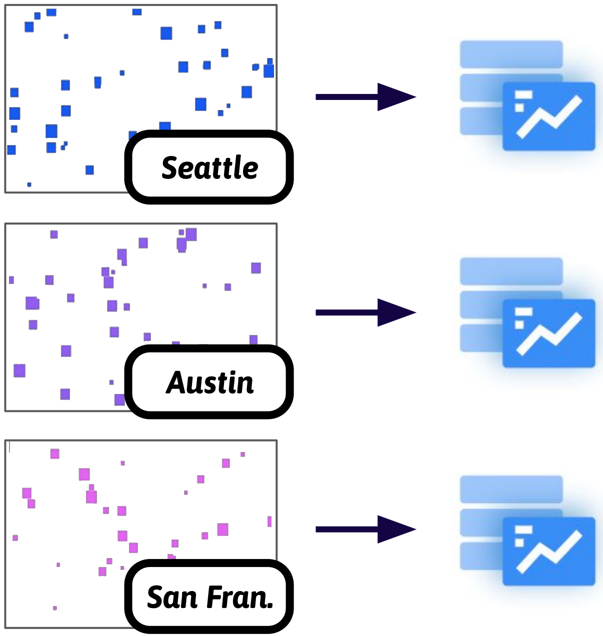
Production ML data is often voluminous, dynamic, and increasingly in the form of streaming data



Complexities of (1) scale and (2) streaming data

Many practitioners try simple sampling techniques; others slice data into segments based on time and other characteristics before conducting analysis





Comparing static windowing, sampling, and profiling

Median and quantile calculation include the following popular approaches:

Static metrics on subsets of data

Predetermine important metrics and store only that information

Random sampling

Store a random sample of the data for further analysis

Data profiling for streaming data

Advanced data structures and algorithms for summarizing data and error

Capturing simple pre-selected metrics for ML data…

```
metrics: {
    mean: 8.0,
    standard_deviation: 1.24,
    quantile_0.25: 5.2,
    ...,
    accuracy: 0.89,
    precision: 0.75,
    recall: 0.92,
}
```

Static metrics approach

Pros:

Fast access to key metrics

Low storage size

Actual metrics on single batch

Cons:

Requires metric pre-selection

Non-mergeable

… isn't enough for root causing production systems!

Using simple pre-selected metrics alone, you can not answer the following:

Est. value of new metric *x* on prior data?

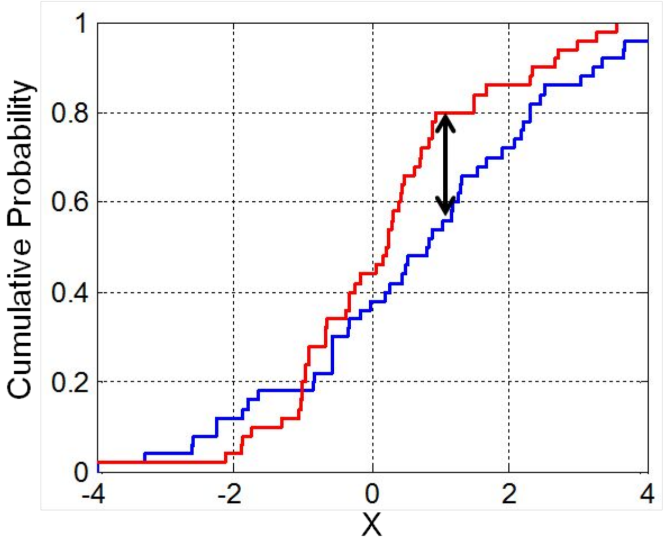
Est, overlap of data with set {*a*, *b*, *c*}?

Relative rank of value *x* on last year's data?

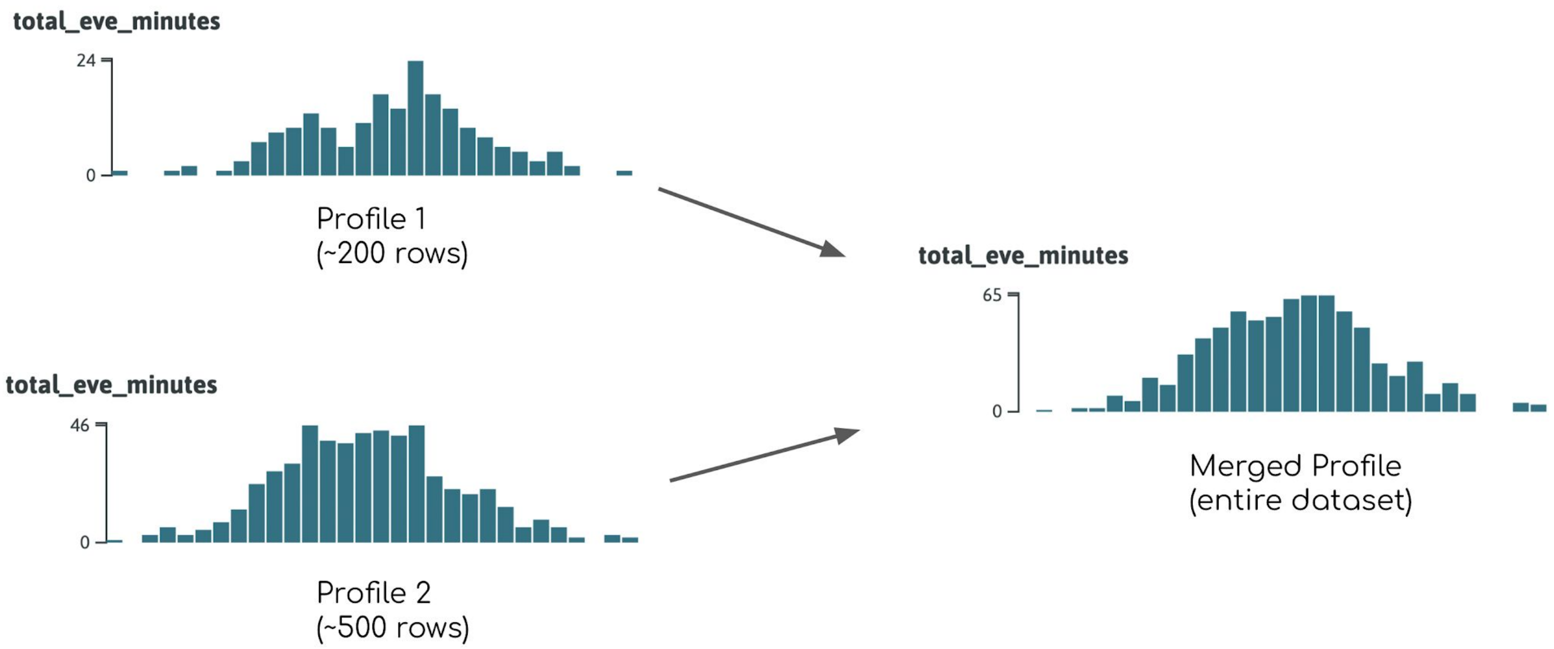
Distribution drift between two datasets?

Error bounds of estimates over the last month of data?

…and many more.



Data mergeability is critical for observing the long tail and rare events



Randomly sampling ML data has issues as well.

```
Sampled rows: 495K               Total rows: 198MM

0 Transaction ID,Customer ID,Quantity,Item Price,
  Total Tax,Total Amount,Store Type,Product
  Category,Product Subcategory,Gender,City Code,
  Age at Transaction Date,Transaction Type,
  Transaction Week,Transaction Batch
1 T24951240379,C267987,12,19.1,24.066000000000003,
  1306.85256,e-Shop,Electronics,Personal
  Appliances,M,9.0,24.0,Purchase,0,2
2 T54251889351,C267740,-3,54.2,17.073,-927.11268
  00000001,MBR,Books,Non-Fiction,M,2.0,36.0,Cancel
  lation,0,2
...
```

Random sampling

Pros:

Same format as original data

High flexibility

Batch or streaming data

Mergeable

Cons:

Poor estimates on tail/outliers

Poor precision (based on %)

High storage size

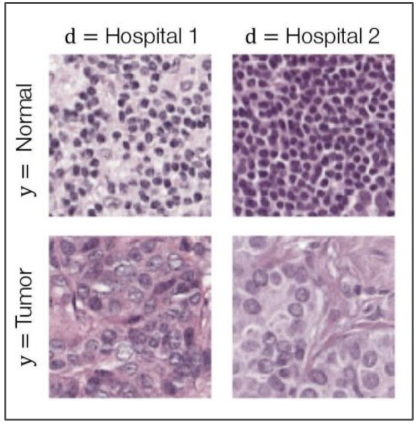
What is data profiling?

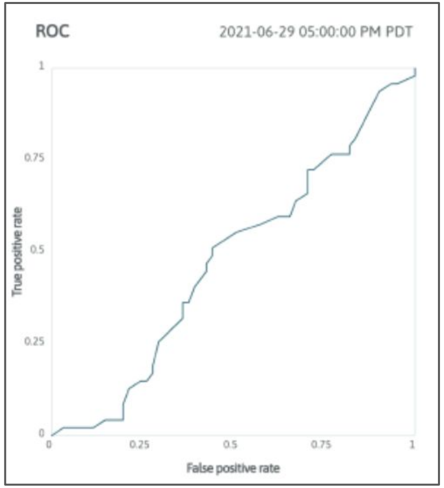
Data profiling is the act of reviewing and analyzing datasets to understand their structure and information. Data profiles can include the following:

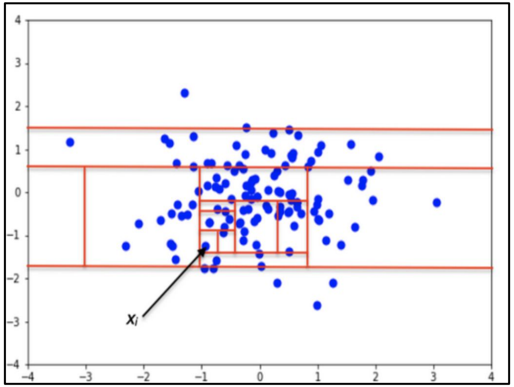
- Collection of descriptive statistics
- Identify different data structures, types, and patterns
- Employ keywords, categorize datasets, and create descriptions
- Conduct data quality examinations
- … and more.

Source: Hanh Truong, “What is Data Profiling?”

Data profiling can include static metrics, but can also contain many more advanced tools needed for analysis

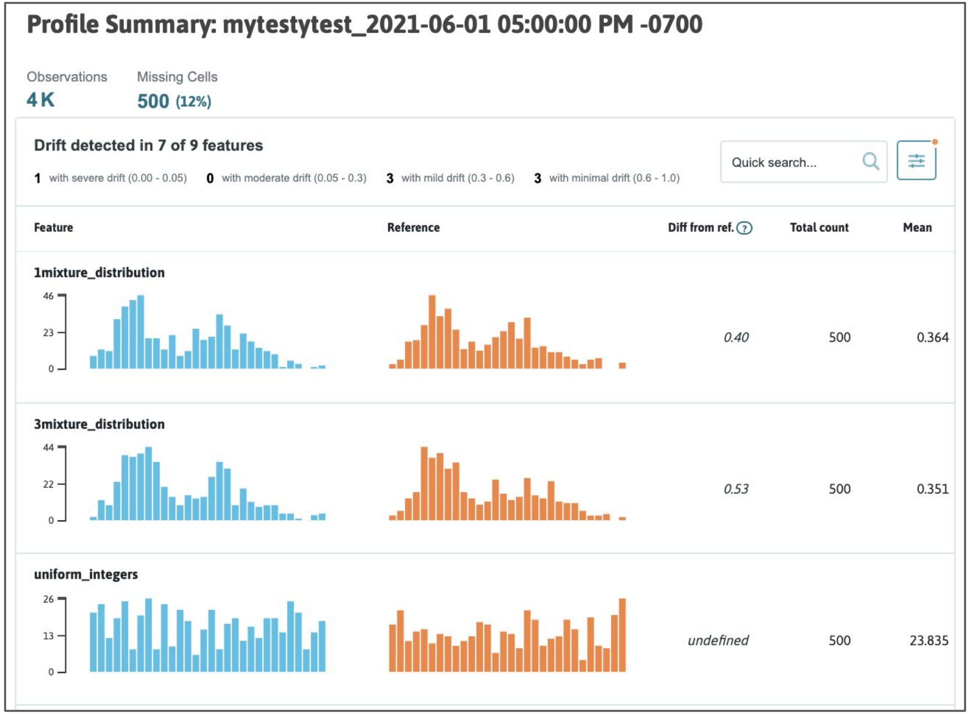






E.g., error bounds for estimates, feature importance, outlier detection, surrogate models.

Sketch-based data profiling for ML data



Data profiling approach

Pros:

Fast access to key metrics

High flexibility

Low memory and storage size

Mergeable

Built on peer-reviewed algos

Cons:

Requires some pre-selection

Underlying algorithm complexity

Building a profiling standard for ML data

Properties of sketch-supported profiling for logging, analysis, and monitoring of ML systems:

- **Lightweight**
- **Configurable**
- **Mergeable**
- **Streaming**
- **Statistically sound**

Powered by:

Apache® DataSketches™

How it works: Notation for median and quantiles

For a stream of numbers $x_1, x_2, \ldots$
with current stream length N:

Rank, $rank(x)$

Number of elements $\leq x$

Relative rank, $r(x)$

Normalized rank, $\dfrac{rank(x)}{N}$

Quantile, $quantile(q)$

Value x s.t. $rank(x) = qN$ or equivalently, $r(x) = q$

Median example

Values: 5 4 1 5 6 2

Sorted: 1 2 4 5 5 6

In this example,

$$rank(4) = 3$$
$$r(4) = \frac{3}{6} = 0.5$$
$$quantile(0.5) = 4$$

Calculating quantiles in P passes over data

Exact calculations

Munro-Paterson proved that the lowest amount of space needed to calculate a quantile in P passes over the data is: $\Omega(N^{1/P})$

You'd need to store N data points to calculate the quantile exactly in streaming setting. Not acceptable!

Approximate calculations

Data sketching techniques allow us to calculate approximate quantiles much more efficiently and in one pass, if desired for streaming.

Numerous algorithms, but KLL (what we use in **whylogs**):

For a single quantile: $(1/\epsilon) loglog^2(1/\epsilon\delta)$

For all quantiles: $(1/\epsilon) loglog^2(1/\delta)$

A brief look at how quantile sketches (KLL) are made

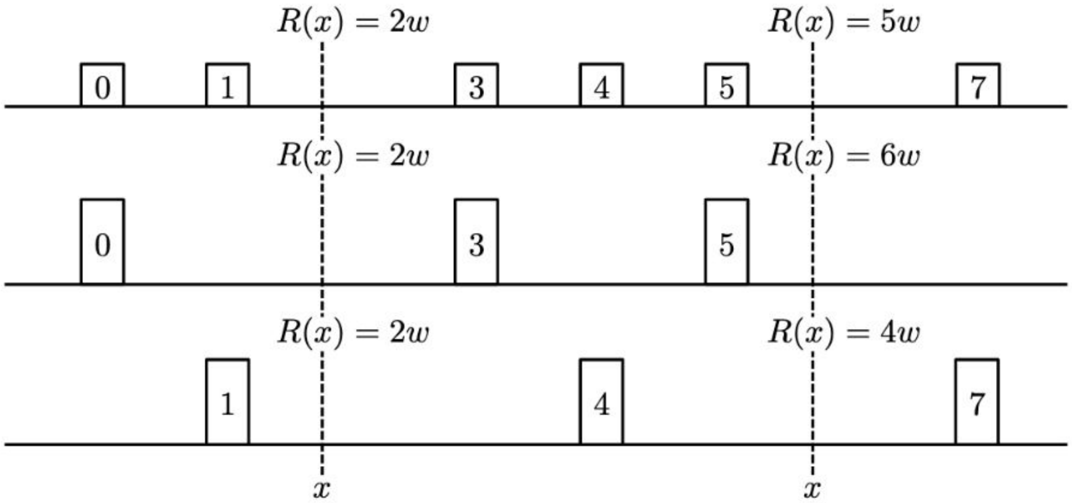


Figure 1: An illustration of a single compactor with 6 items performing a single compaction operation. The rank of a query remains unchanged if its rank with in the compactor is even. If it is odd, its rank is increased or decreased by w with equal probability by the compaction operation.

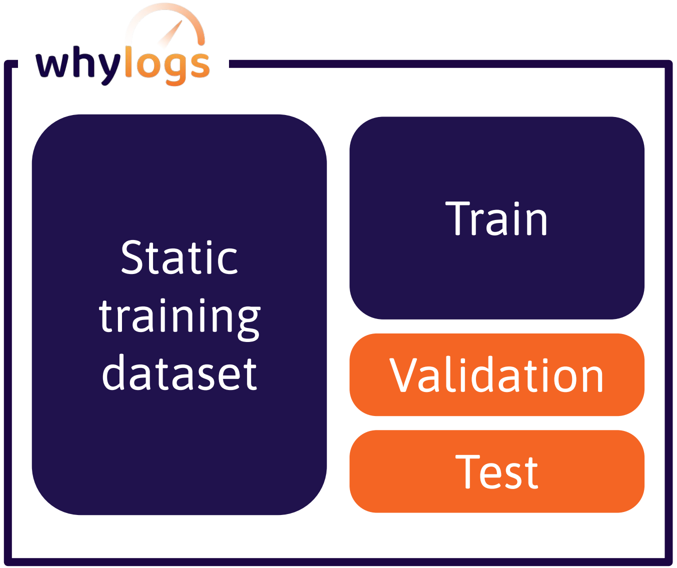
Source: Cardin, Lang and Liberty 2016

Considerations for the whylogs library

Properties of profiling that make whylogs great for logging, analysis, and monitoring ML systems:

- **Lightweight**
- **Mergeable**
- **Configurable**
- **Streaming**
- **Statistically sound**

Profiling training data and other static datasets



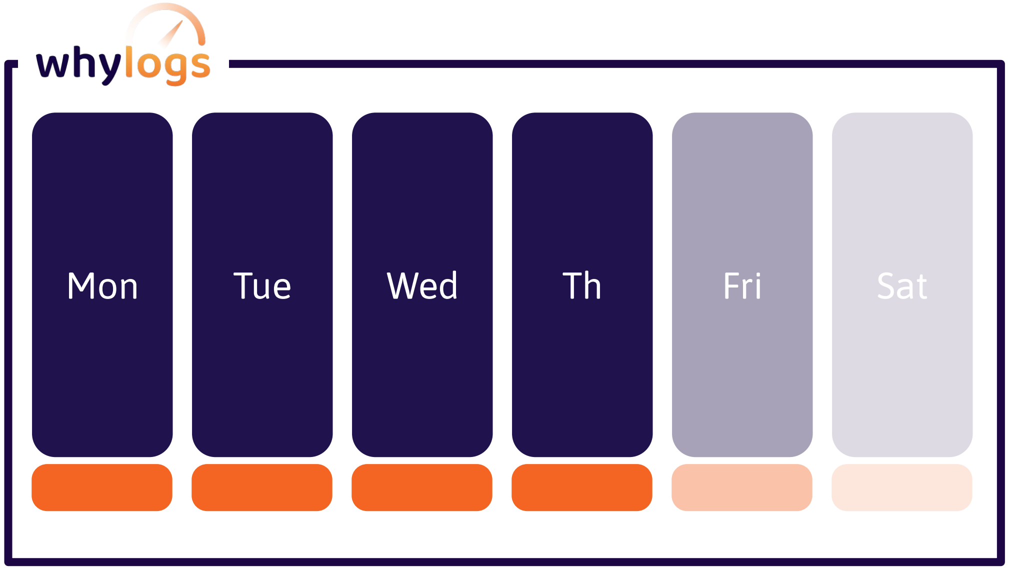
Profile static datasets such as training datasets to store, analyze, and use as a comparison for monitoring.

Uses the same calculations as other profiling, so emphasis on lightweight, speed, and common use cases.

Profiling ongoing production data

Most typical use case, profiling batch or streaming production data.

The underlying data (and perhaps actuals for performance metrics) gets logged regularly while you serve production traffic.



Single profile analysis, but added value for 2+ profiles

whylogs

	Single profile	Two profiles	Three or more
Data documentation	✓	✓	✓
Exploratory data analysis	✓	✓	✓
Data unit testing	✓ NEW!	✓	✓
Ad-hoc comparison to Baseline		✓	✓
Continuous monitoring			✓

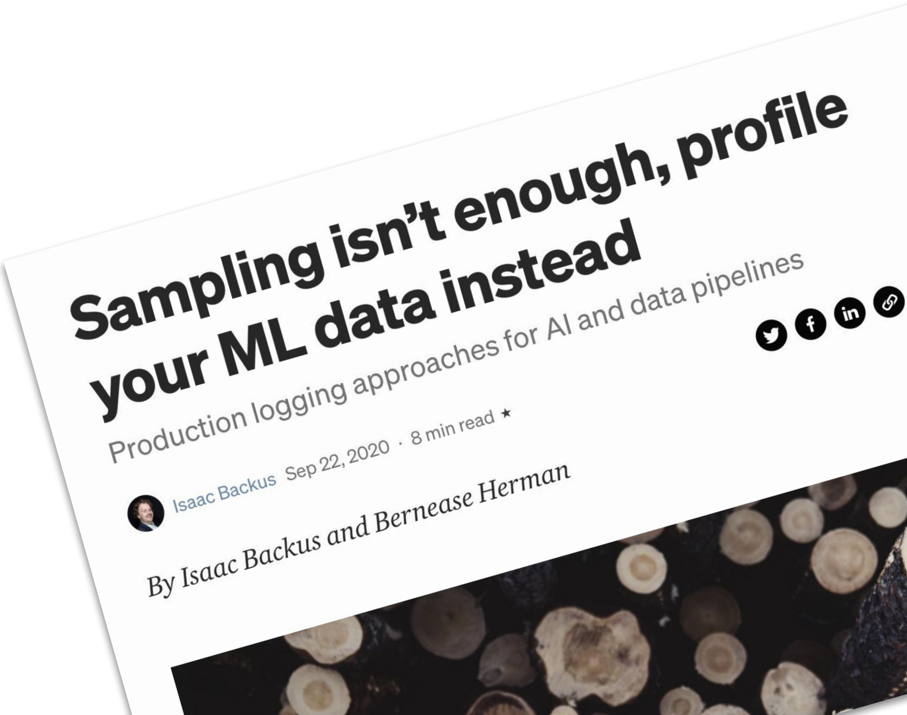
With multiple data profiles, powerful analyses like drift detection, event monitoring, and automated data unit testing become available.

Data sampling versus profiling experiments: Comparing error on common statistical distributions

Experimental procedure:

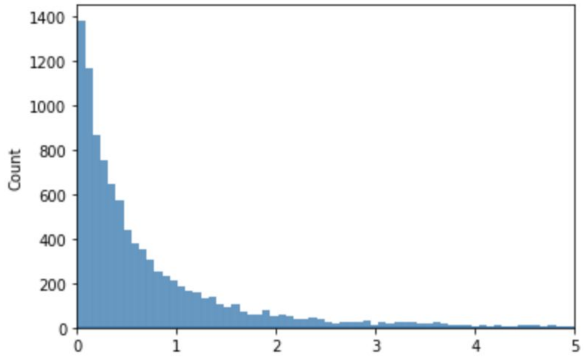
For each statistical distribution:

1. Randomly sample 10^5 records
2. Sample a subset of `n_sample` records such that the subset is as many bytes as the profile. This is to compare apples to apples.
3. Compare with exact value on sample
4. Repeat steps 2 through 4 for a total of 24 runs and average the results



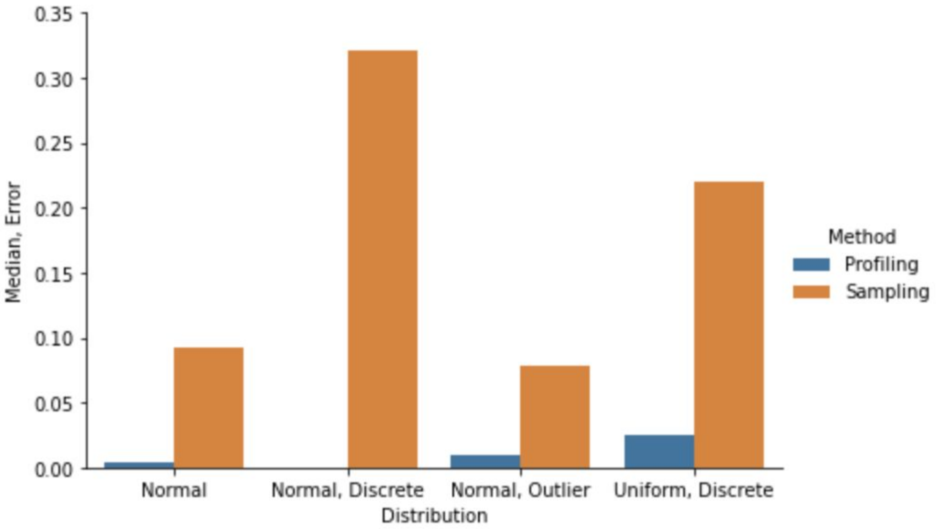
Data sampling versus profiling experiments: Statistical distributions chosen for experiments

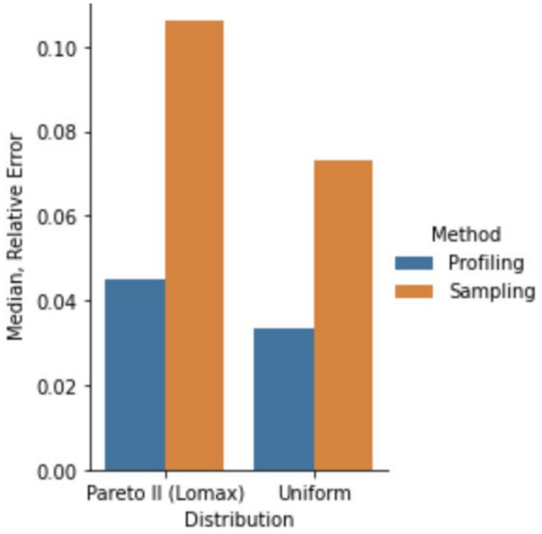
Distribution	Parameters	Purpose
Normal	mu = 0, std dev = 1	A broad class of data. Unskewed, has a tail but is peaked around the center
Uniform	min = 0, max = 1	Data without a tail that is evenly sampled across its domain.
Pareto (type II)	shape = 2, min = 0	A broad class of skewed data with a long tail/outliers.
Discretized normal	mu = 0, std dev = 1 discretized into ~10 categories	Non-uniformly sampled categorical data, occasionally with outliers
Discretized pareto (type II)	shape = 2, min = 0 discretized into ~10 categories	Very non-uniformly sampled categories, with rare events/outliers.
Discrete Uniform	min = 0, max = 1 10 categories	Evenly sampled categorical data



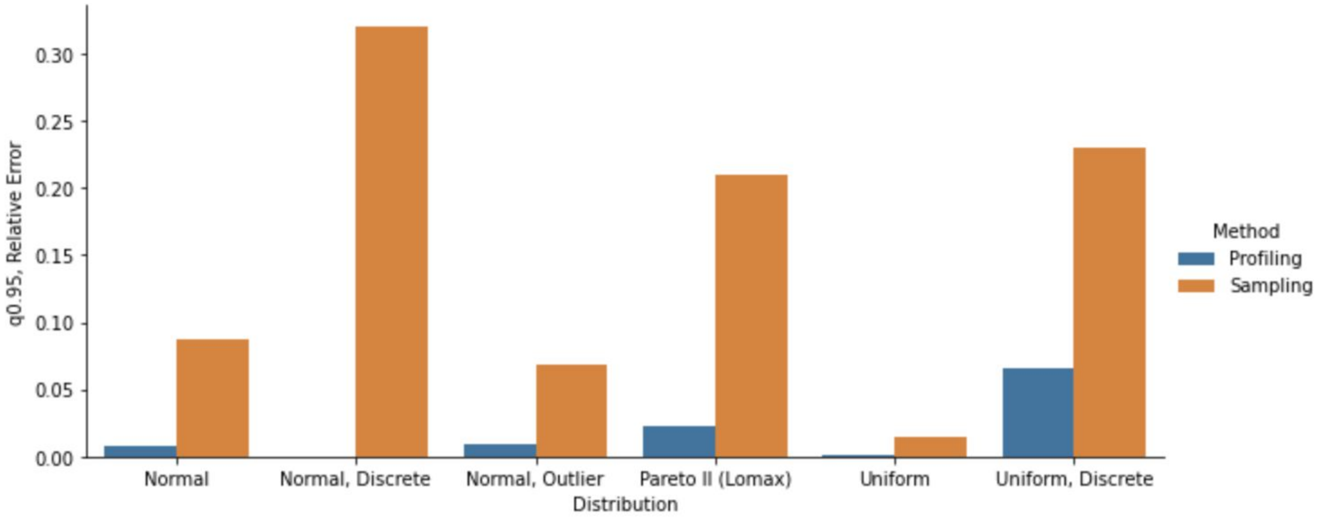
Pareto Type II, or Lomax distribution

Data sampling versus profiling experiments: Comparing error on median across distributions

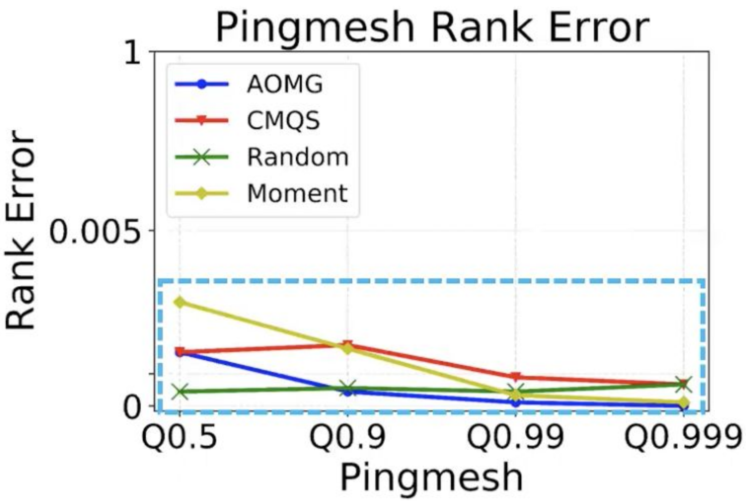


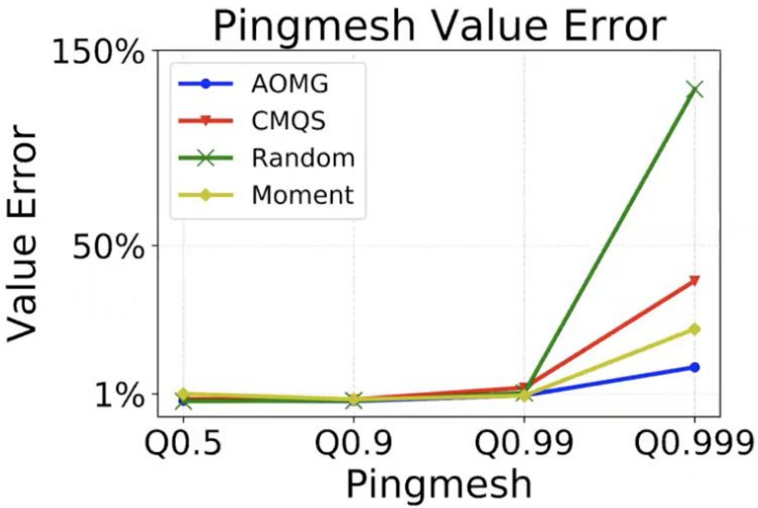


Data sampling versus profiling experiments: Comparing error of across q0.95 across distributions



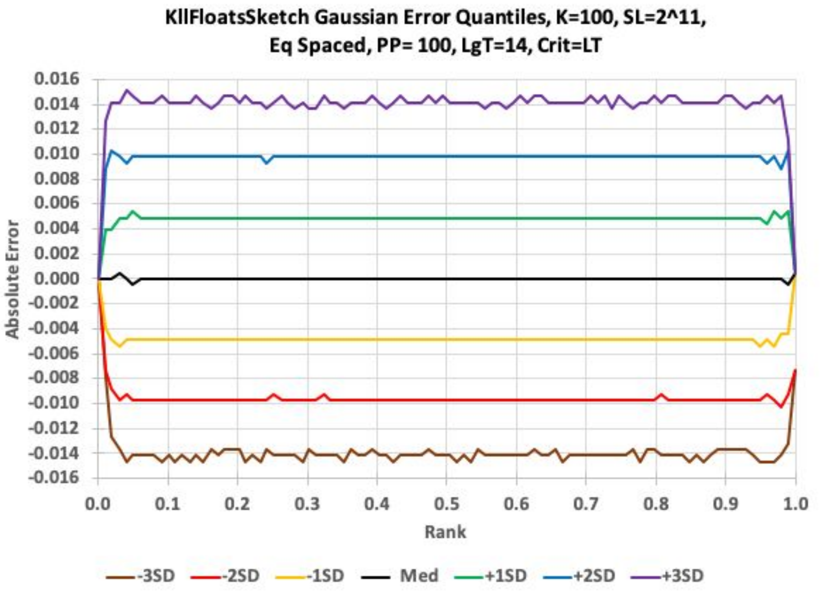
But even low rank error can have a large effect on the tail of the distribution where values may be high

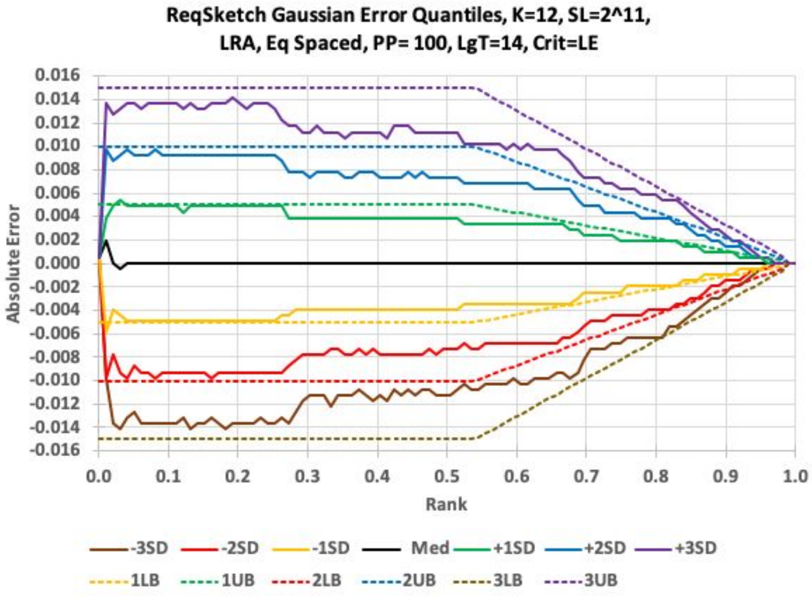




Source: Gangmuk Lim, ICSE 2020 Presentation

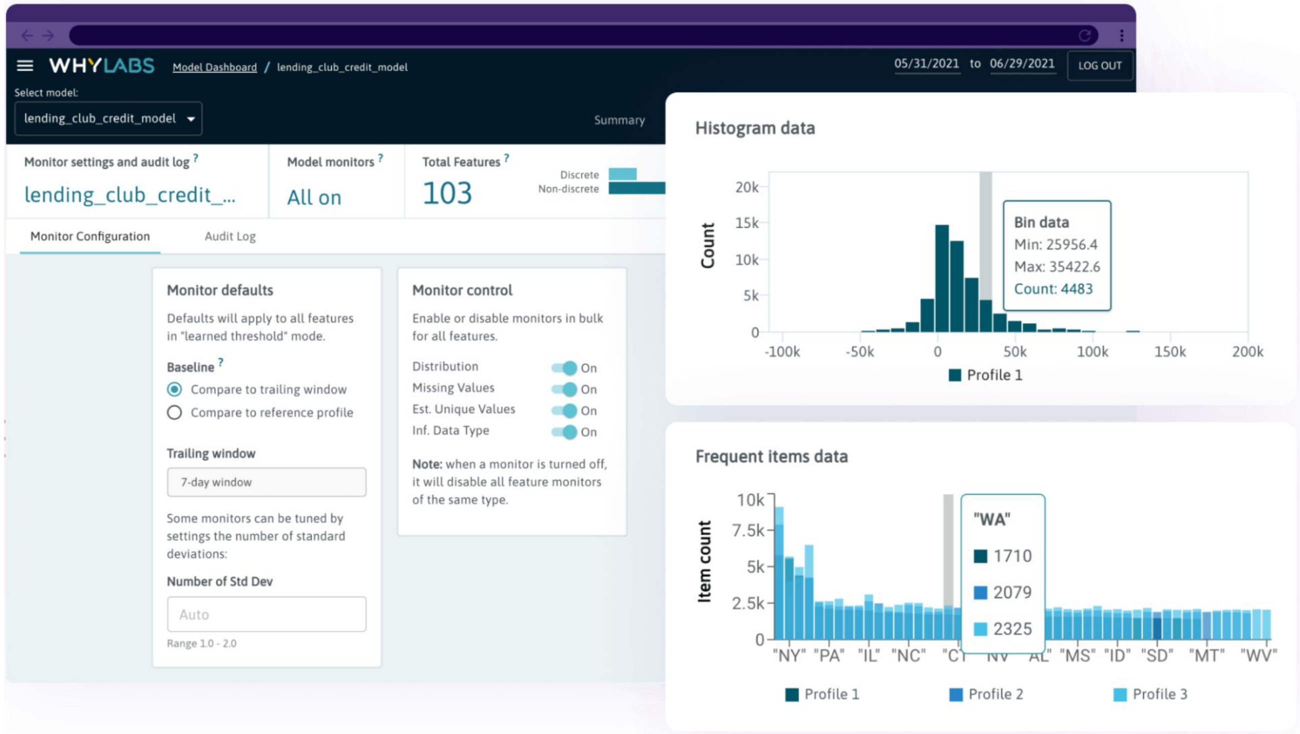
Current sketch treats error evenly across rank, but opportunities to prioritize left or right tail of data





Source: Apache DataSketches, Relative Error Quantiles (REQ)

Want to extend functionality beyond open-source whylogs profiles? Try the WhyLabs SaaS platform



Thank you!
Questions?

Also, help build the open standard for data logging:

github.com/whylabs/whylogs

join.slack.whylabs.ai

Contact me:

In-person at Data Council Austin

Email: **bernease@whylabs.ai**

Social media: **@bernease**

Instructions for getting WhyLabs swag:

- Star the **whylogs** project on Github
- Join our **Community Slack**
- Submit a **form** with relevant info at bit.ly/whylogsswag

A subset of ML issues encountered in production

- Experiment/production environment mismatch
- Wrong model version deployed
- Underprovisioned hardware
- Inappropriate hardware
- Latency/SLA issues
- Data permissions misconfigured
- Untracked changes broke prod
- Traffic sent to the wrong model
- Computational instability
- Customers gaming the model
- PII data exposed
- Expected accuracy doesn't materialize

- Pre-processing mismatch in experiments vs. production
- Retrained on faulty data
- Accuracy improves on one segment, regresses in others
- Outliers predicted incorrectly
- Bias identified
- Correlation with protected features
- Overfitting on training/test
- Surge in missing values
- Surge in duplicates

- Poor performance on outliers
- Data quality issues affect accuracy
- Production data doesn't match test/training
- Accuracy is decaying over time
- Data drift in inputs
- Concept drift in outputs
- Extreme predictions for out of distribution data
- Model not generalizing on new data / new segments
- Major consumer behavior shift

...or it simply doesn't work, and nobody knows why!

Most ML issues are observable from the data itself

- Experiment/production environment mismatch
- Wrong model version deployed
- Underprovisioned hardware
- Inappropriate hardware
- Latency/SLA issues
- Data permissions misconfigured
- Untracked changes broke prod
- Traffic sent to the wrong model
- Computational instability
- Customers gaming the model
- PII data exposed
- Expected accuracy doesn't materialize

- Pre-processing mismatch in experiments vs. production
- Retrained on faulty data
- Accuracy improves on one segment, regresses in others
- Outliers predicted incorrectly
- Bias identified
- Correlation with protected features
- Overfitting on training/test
- Surge in missing values
- Surge in duplicates

- Poor performance on outliers
- Data quality issues affect accuracy
- Production data doesn't match test/training
- Accuracy is decaying over time
- Data drift in inputs
- Concept drift in outputs
- Extreme predictions for out of distribution data
- Model not generalizing on new data / new segments
- Major consumer behavior shift