Mathscapes®

Anomaly scoring before the baseline update

An anomaly score depends on which observations define its baseline. This research interface makes the state, scoring order and threshold visible.

Anomaly inspector

Research concept

Historical state

Event	Value	Running mean
01	9	9
02	10	9.5
03	11	10
04	10	10

n = 4 · mean 10 · variance 0.5

Incoming event 05: value 14

Score before update

5.66

Review at +3

Include 14 first

1.86

Below +3

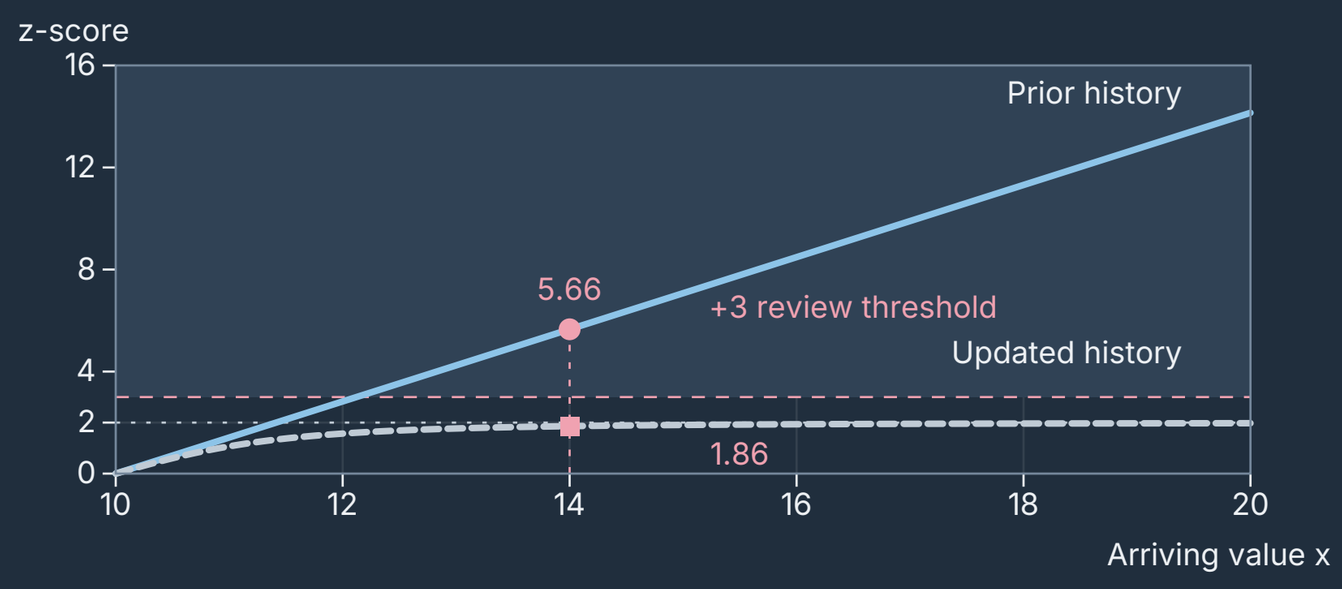
Same arrival; a different reference changes its score.

```
source.streamingZScore().take(100).toArray()
```

Illustrative interface; +3 is not validated. Iterflow · DOI 10.5281/zenodo.18610143

Baseline sensitivity

History [9, 10, 11, 10]; only the arrival changes.



Computed example · DOI 10.5281/zenodo.18610143

Updated baseline

Updated score < 2

For every finite x > 10

Effect of including the arrival

Including the arrival shifts the mean and expands the spread. The event appears closer to a baseline that now contains it.

The +3 threshold is illustrative, not a validated operating point.

Effect of scoring order

The same arrival, 14, scores 5.66 against past data but 1.86 after inclusion. This worked example does not validate an anomaly detector.

For d = x - 10, the updated score is 2d / sqrt(d² + 2.5). Its limit is 2; the prior score is unbounded. Neither is a probability.

Welford updates after scoring; two initial outputs are NaN. Zero spread and missing values need policies.

Implementation and benchmarks

Iterflow composes lazy transforms and online statistics. Scalar estimators use O(1) state; windows O(k), collectors and median O(n). Native helpers already cover transforms.

Manuscript Node 22/ARM64 throughput reports lazy filtering at 7,818 vs 26 operations/s against arrays, but z-score at 175 vs 594 against a loop. Composition can avoid work while adding overhead. These benchmarks were not rerun.