Know Your Data: The stats behind the numbers

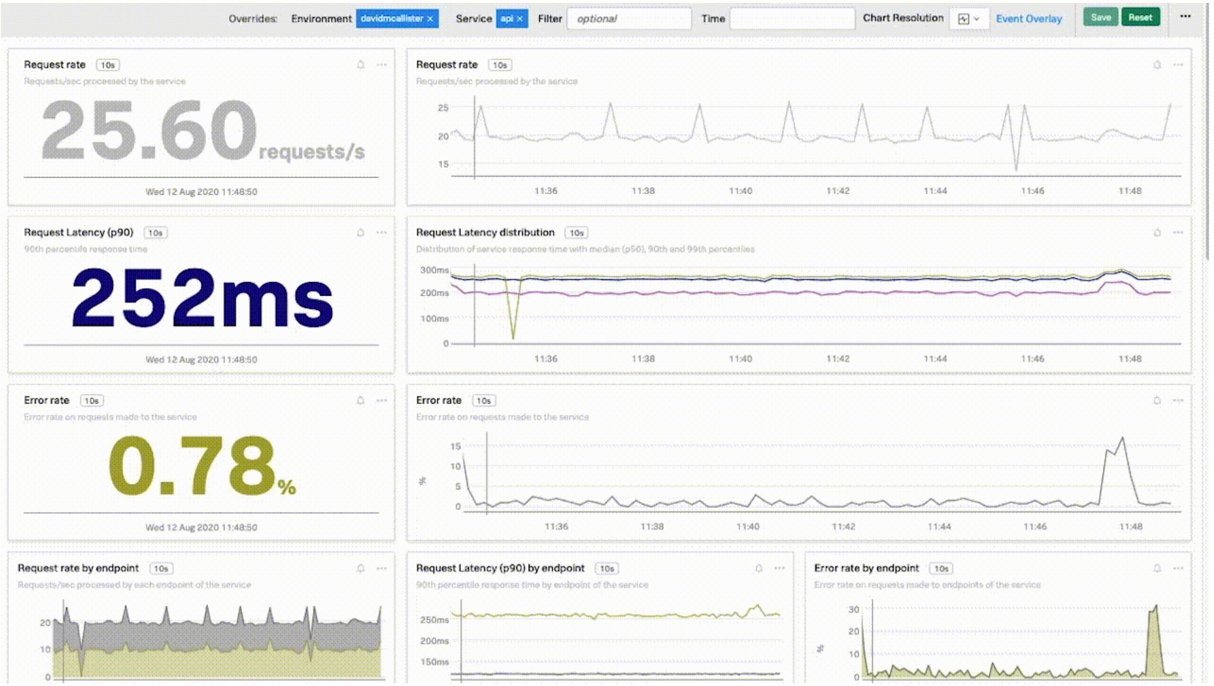
Dave McAllister

Sr. OSS Technologist

NGINX

Quick:
What’s the difference between Mean, Median and Mode?

Monitoring is a numbers game



- Metrics are numbers that represent selected behavior
- Generally
  - Timestamped
  - Key-Values
- Data, to be useful, must be
  - Aggregated
  - Analyzed
  - Visualized

Some questions to ponder

- How do you deal with outliers (spikes) in monitoring?
- How do you get a representative value between vastly different quantities (rates, speeds)?
- How do you arrive at values to represent rate of change over time?

Mean, Median, Mode

Data:

2, 6, 4, 9, 5, 1, 7, 8, 1, 9, 9, 1, 10, 2, 9, 6, 7, 2, 1, 4, 7, 1, 10, 9, 2, 7, 1, 1, 4, 3, 5, 6, 3, 8, 1, 8, 4, 7, 6, 3, 9, 9, 9, 4, 9, 1, 4, 1, 9, 8, 10, 10, 1, 1, 1, 7, 10, 9, 7, 3, 7, 4

Mean:

A measure of central tendency that represents the average value of a set of data.

Mean = 5.444

Median:

Represents the middle value in a set of ordered data

Median = 6

Mode:

The value that appears most frequently in a set of data.

Mode = 1

Mean, Median, Mode

Data:

2, 6, 4, 9, 5, 1, 7, 8, 1, 9, 9, 1, 10, 2, 9, 6, 7, 2, 1, 4, 7, 1, 10, 9, 2, 7, 1, 1, 4, 3, 5, 6, 3, 8, 1, 8, 4, 7, 6, 3, 9, 9, 9, 4, 9, 1, 4, 1, 9, 8, 10, 10, 1, 1, 1, 7, 10, 9, 7, 3, 7, 4

Mean:

A measure of central tendency that represents the average value of a set of data.

Median:

Represents the middle value in a set of ordered data

Median = 6

Mode:

The value that appears most frequently in a set of data.

Mode = 1

Mean = 5.444
Or is it 4.130 or 2.791?

Means to an End

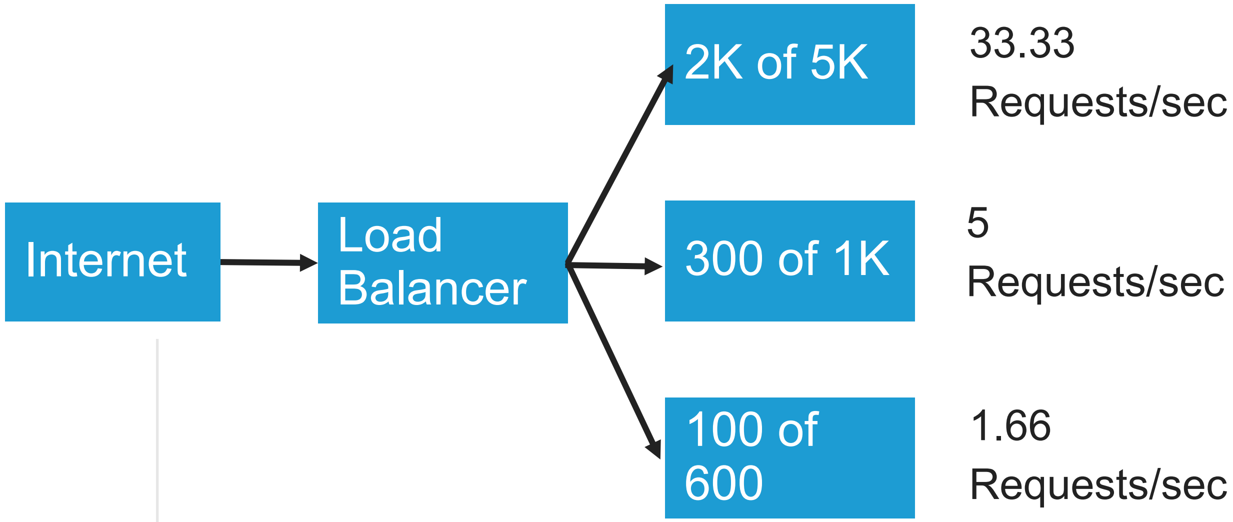
Arithmetic, Harmonic, Geometric, Trimmed, Weighted, Moving

- Each has potential uses and drawbacks
- Often already implemented in the monitoring software
- Can give very different results
- Can make like and unlike comparisons easier

Arithmetic

- Most common
- Is the central point in a normal distribution
  - This is not the 50% mark (mostly)
- Useful for comparing current to previous conditions
- May be aggregated into groups (time series)

In a time series, we usually calculate constantly to incorporate new data

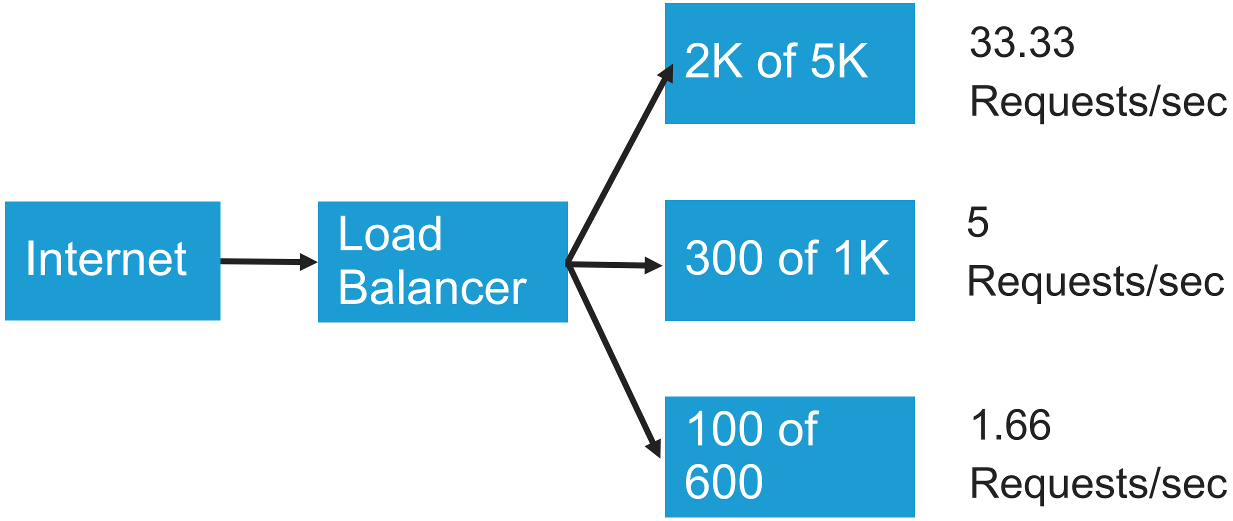


Amean = (33.33 + 5 + 1.66) / 3

Amean = 13.33 Requests per second

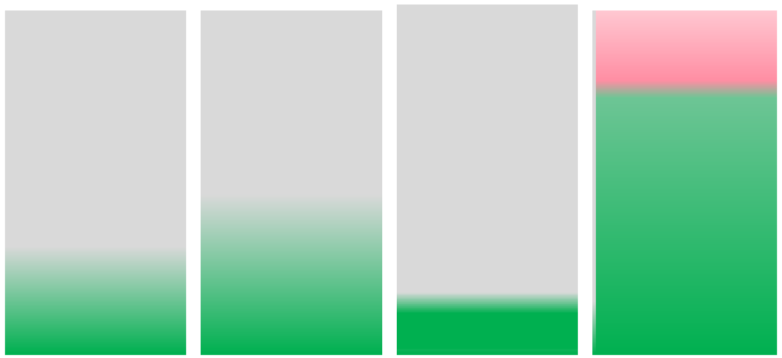
Geometric

- Another central tendency
- Often used for things growing exponentially
- Multiply all the items together, take the nth root
- In DevOps
  - Average number of deploys per unit of time
  - Average lead time for changes
  - MTTR
  - Throughput



Geometric mean = (33.33 * 5 * 1.66) ^ (1/3) = **6.525 requests per second**

Harmonic



- Measure the performance where multiple systems are involved
- Weights the lowest figure the highest
- Divide n by the sum of the reciprocals
- In DevOps
  - Performance within range
  - Overall indication of latency or thruput
  - Use in complex environments
  - Especially useful for outliers

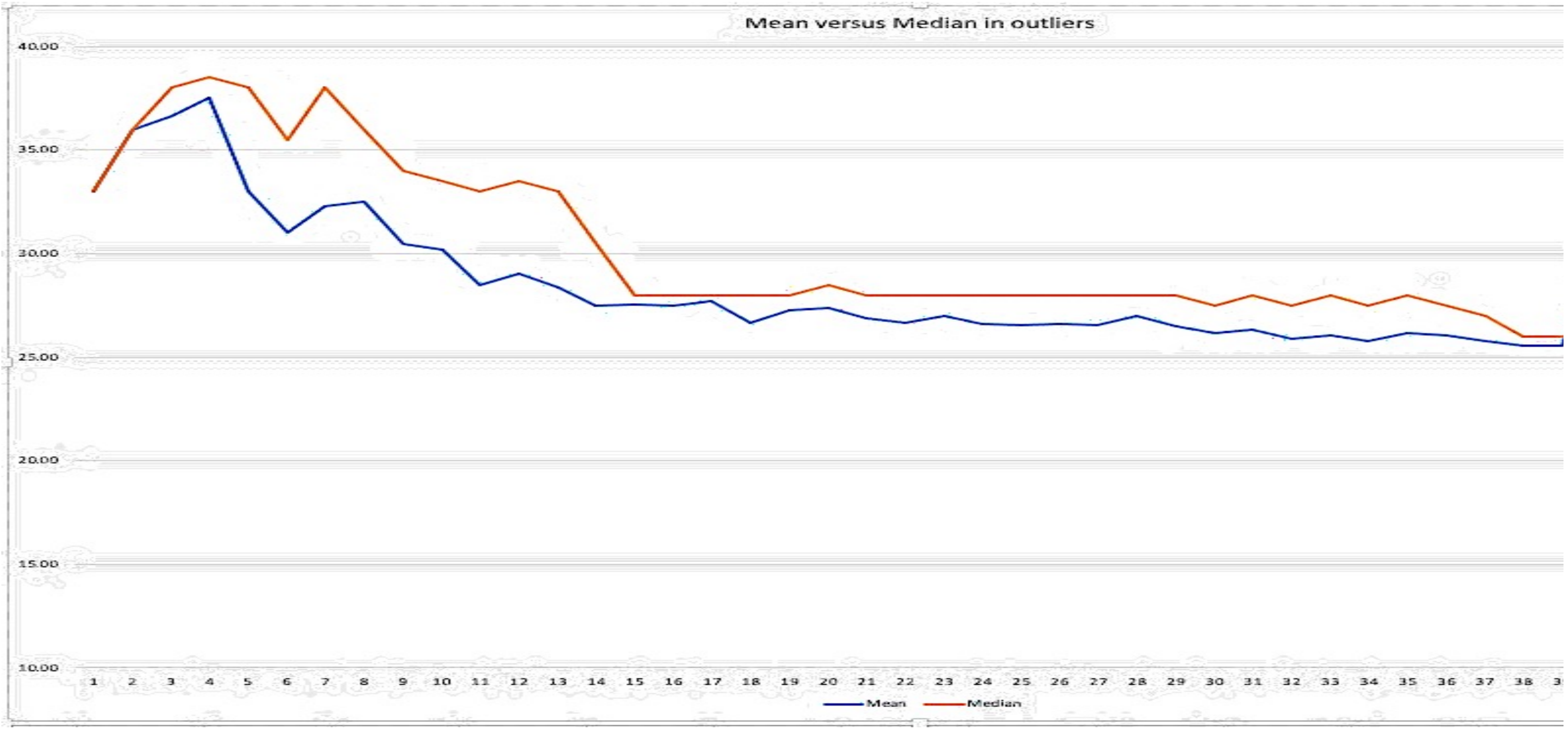
$$n / (1/x1 + 1/x2 + \ldots + 1/xn)$$

	First % used	Second % Used
Node 1	30%	30%
Node 2	40%	40%
Node 3	20%	10%
Node 4	10%	80%
Harmonic	**19.19%**	**32.82%**

Median

- Amazingly underutilized!
- Center value of a sorted list
- Median is always the 50% point of a normal curve

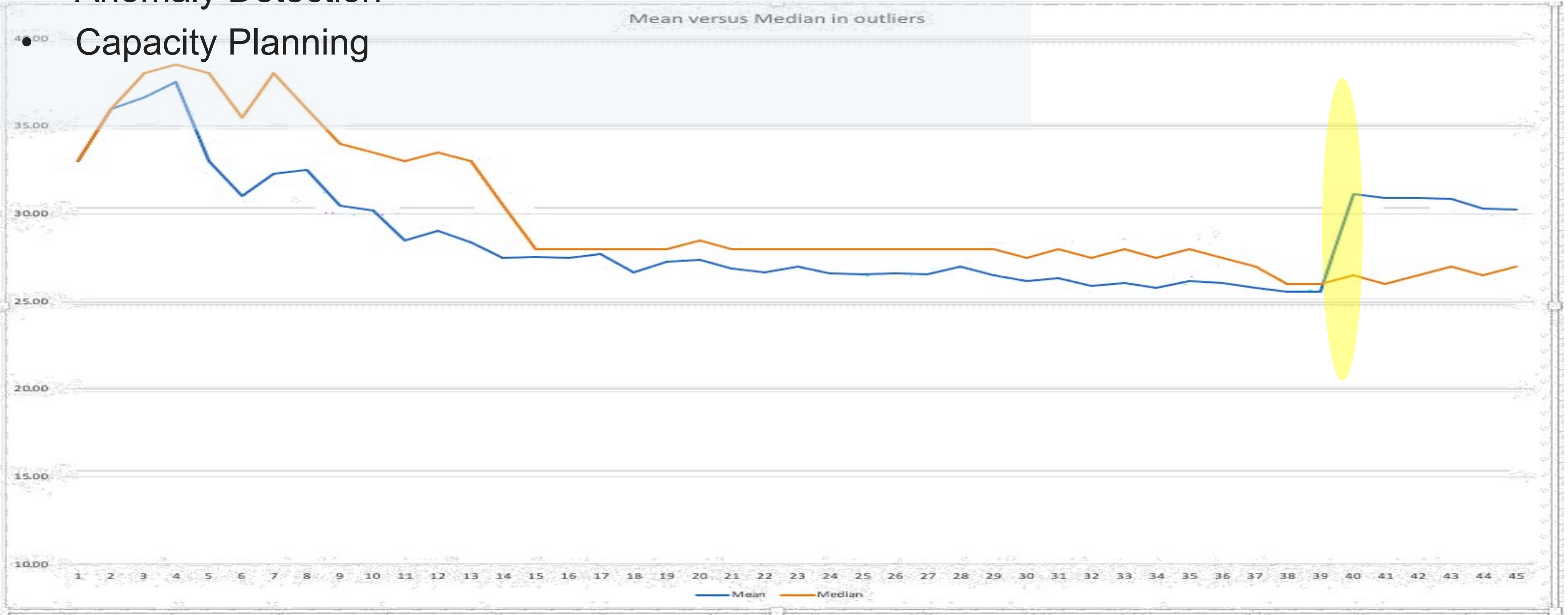
Mean	25.25
Median	26

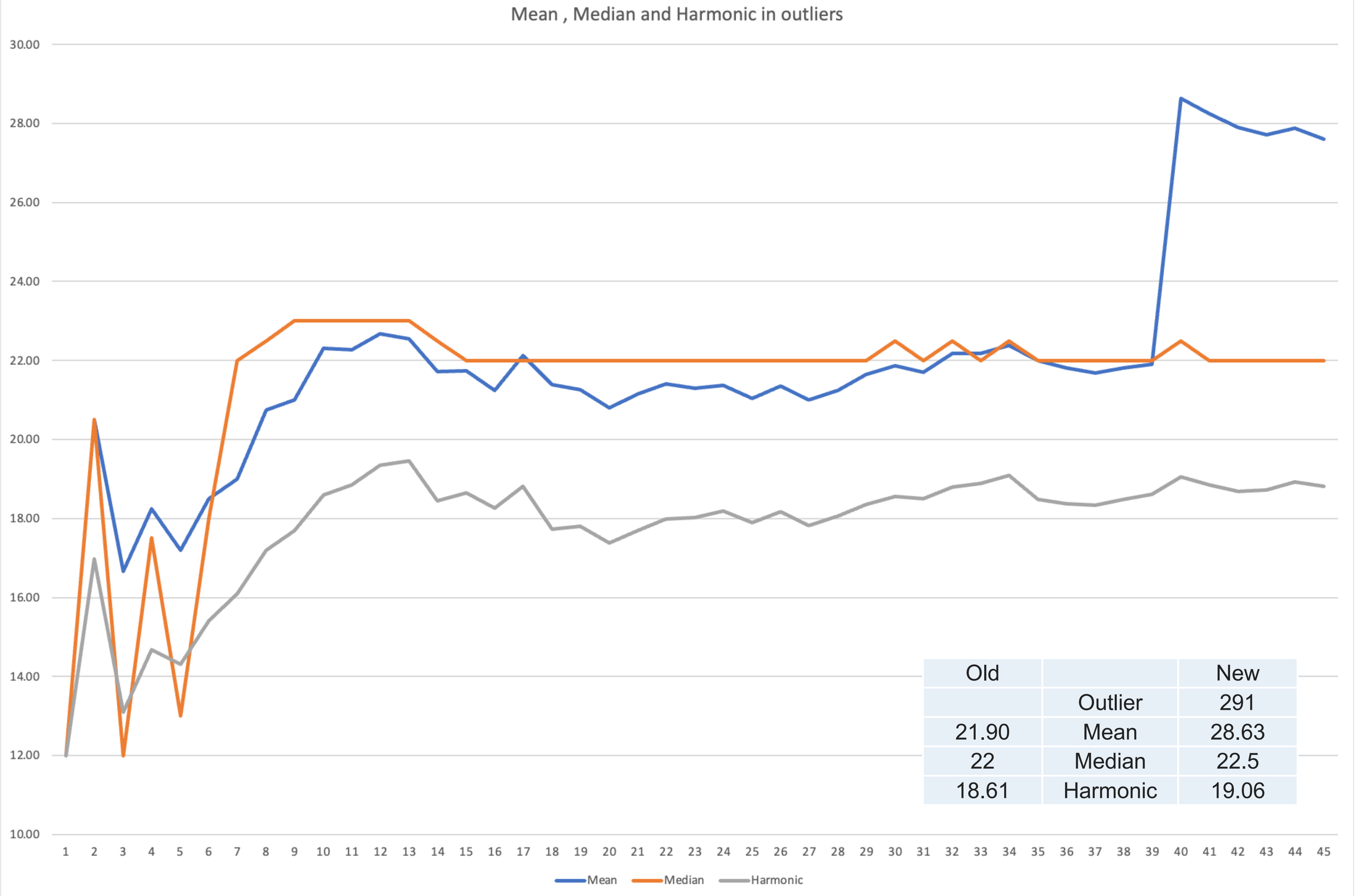


Choosing Between Mean and Median

- Mean can be impacted by outliers
- Resilience is better in median
- In DevOps
  - Response time monitoring
  - Anomaly Detection
  - Capacity Planning

Old		New
	Outlier	250
25.54	Mean	31.15
26	Median	26.5

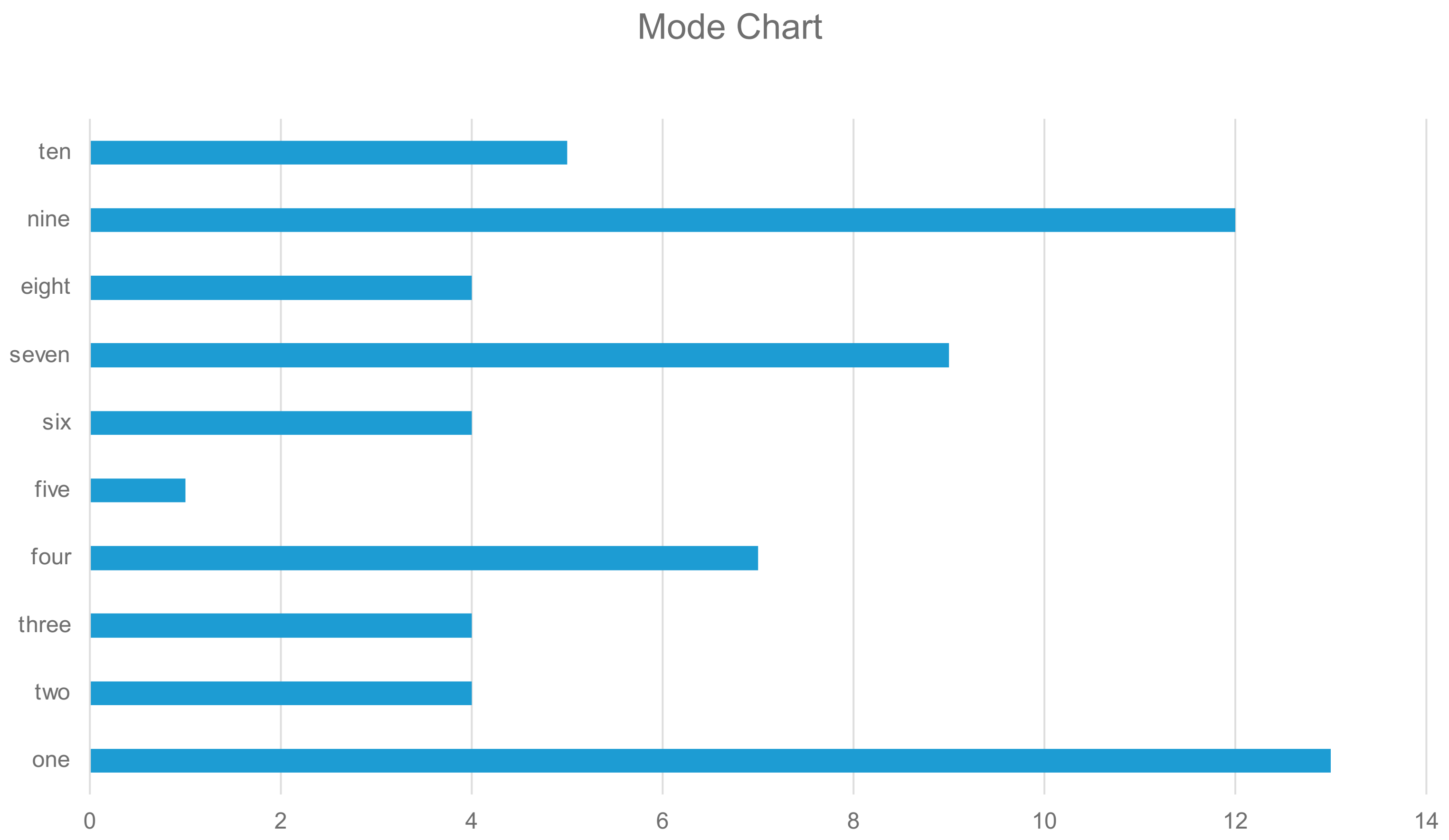




Old		New
	Outlier	291
21.90	Mean	28.63
22	Median	22.5
18.61	Harmonic	19.06

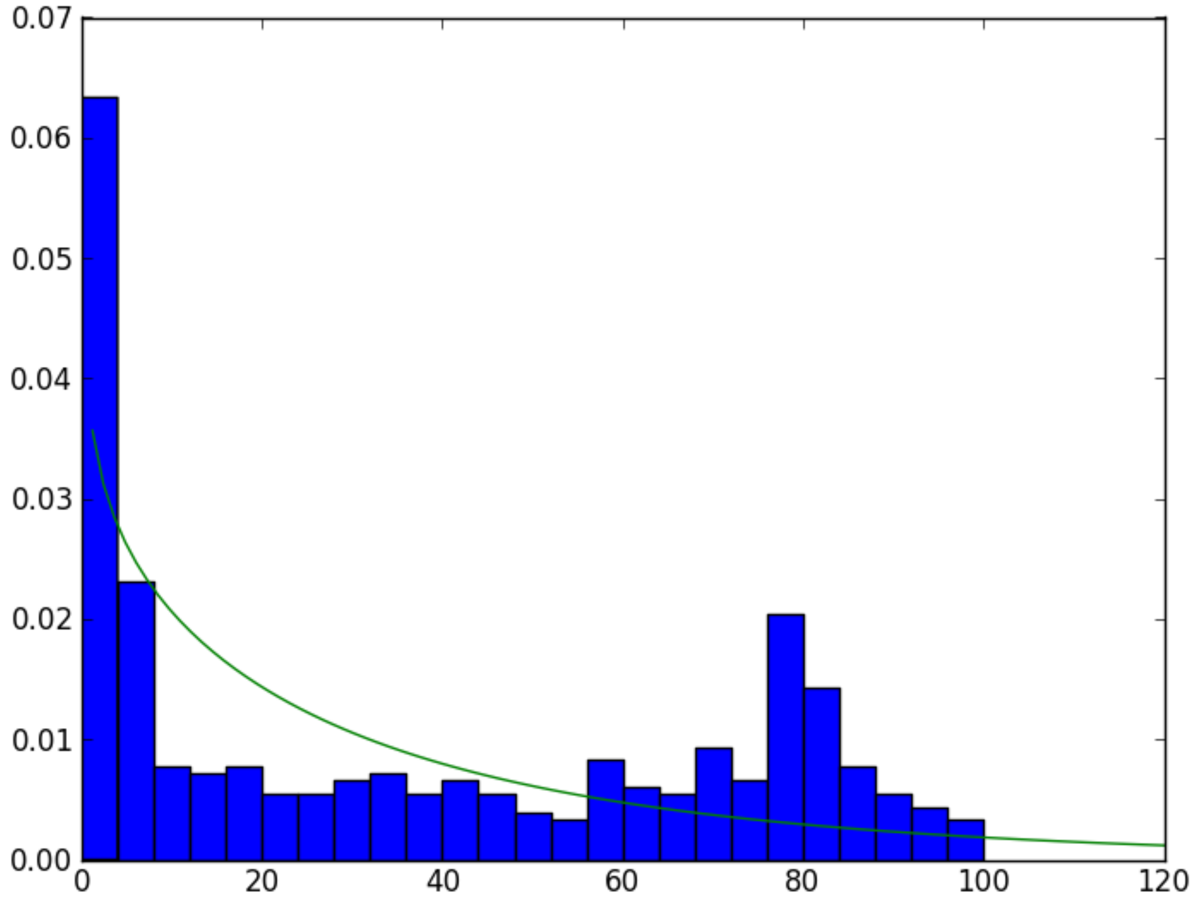
How about the Mode?

- The most commonly recurring value in the set
- Often presented as a histograph
- Not commonly used in DevOps, mostly inferential
  - Log Analysis
  - Security Monitoring
  - User Behavior Analysis



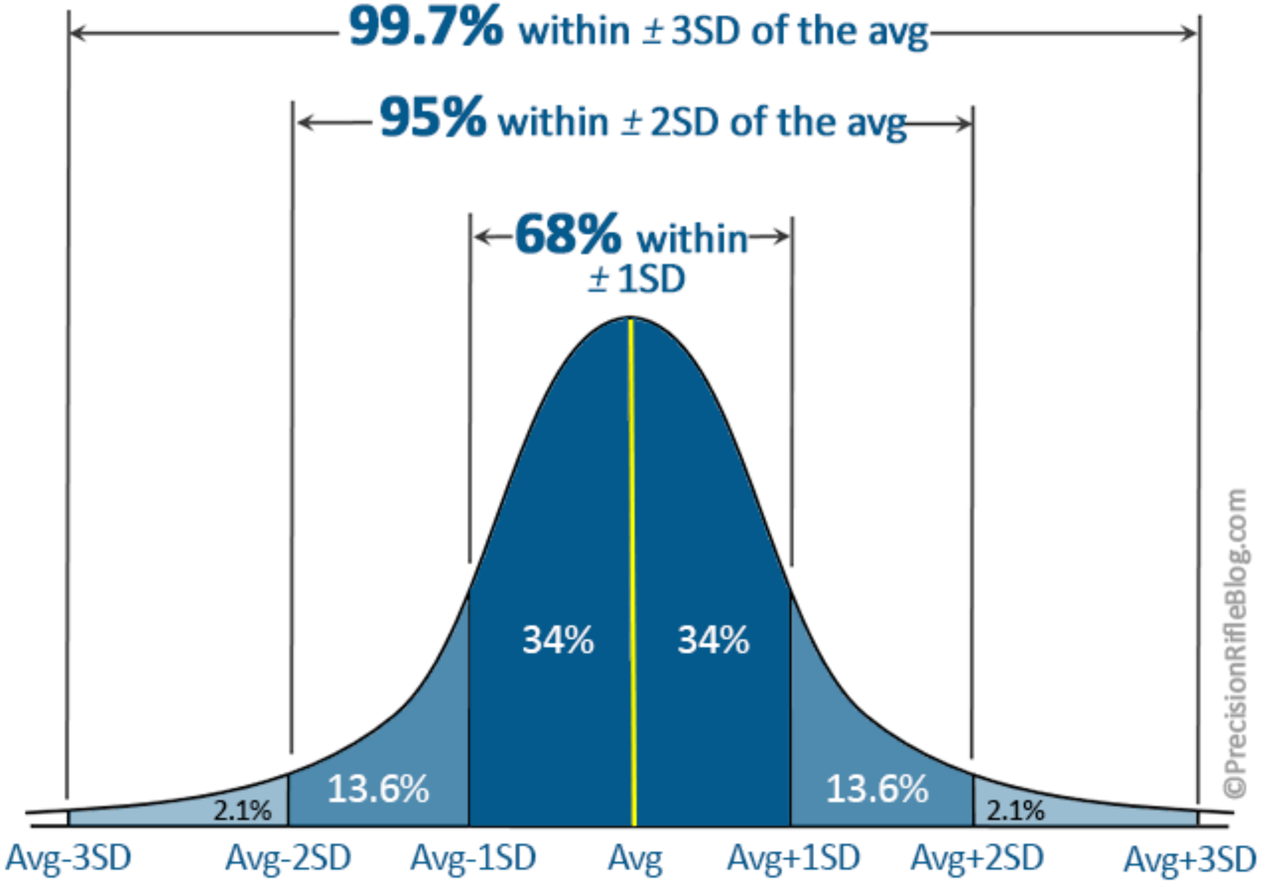
Distributions

- Normal
  - Data equally distributed
- Poisson
  - used to model the occurrence of rare events
- Beta
  - Success/failure of binomial events
- Exponential
  - Time between async events
- Weibull
  - Likelihood of failure
- Log-normal
  - Values based on many small events



Slight sidetrack: Standard deviation

- Measures the variability of your data
- Identifies trends and outliers
- NOT percentage based
  - Except with coefficient of variability
  - CV=Mean / std dev X 100
  - Useful for measurement ignoring range
- SRE cases
  - Lead times
  - Recovery times
  - Anomalies (alerts)
  - SLO / SLI



Deeper dive: Weibull

- Usually used for time-to-failure
- Defined by a Shape and a Scale parameter
  - This can be challenging
  - Don’t ask the math
    - R does it for you!

Component	Time-to-Failure
Spinning Rust	500 hours
Memory	1000 hours
Power Supply	1500 hours
CPU	2000 hours
SSD	2500 hours

```
library(fitdistrplus)
data <- c(500, 1000, 1500, 2000, 2500)
fit.weib <- fitdist(data, "weibull")
summary(fit.weib)
```

```
Fitting of the distribution ' weibull ' by maximum likelihood
Parameters :        estimate        Std. Error
          shape       1.0624082     0.3820112
          scale    2158.2561922   943.0326941
```

```
p.failure <- pweibull(3000, shape = fit.weib$estimate[1],
scale = fit.weib$estimate[2])
1 - p.failure
```

```
[1] 0.2905977
```


Deeper dive: Exponential

User request arrival time	Count
5 seconds	120
10 seconds	60
15 seconds	30
20 seconds	10

- Models the “rate” (time between events that are unrelated)
- Use cases
  - Network performance
  - User Requests
  - Messaging service
  - System failures
- Don’t ask the math
  - R does it for you

```
library(MASS)
data <- c(rep(5, 120), rep(10, 60), rep(15, 30), rep(20, 10))
fit.exp <- fitdistr(data, "exponential")
summary(fit.exp)
```

```
      estimate
rate 0.04232899
```

```
p.request <- pexp(10, rate = fit.exp$estimate)
p.request
```

```
[1] 0.3943056
```

Deeper Dive: Probability Distribution Function

- Describes the probability
  - Of a random variable
  - Shows likelihood of observation
  - Is non-negative
- Make informed decisions

Math:

$f(x) = (1 / (\sigma * \mathrm{sqrt}(2\pi))) * \exp(-((x - \mu)^2 / (2\sigma^2)))$

•f(x) is the PDF at a given value x,

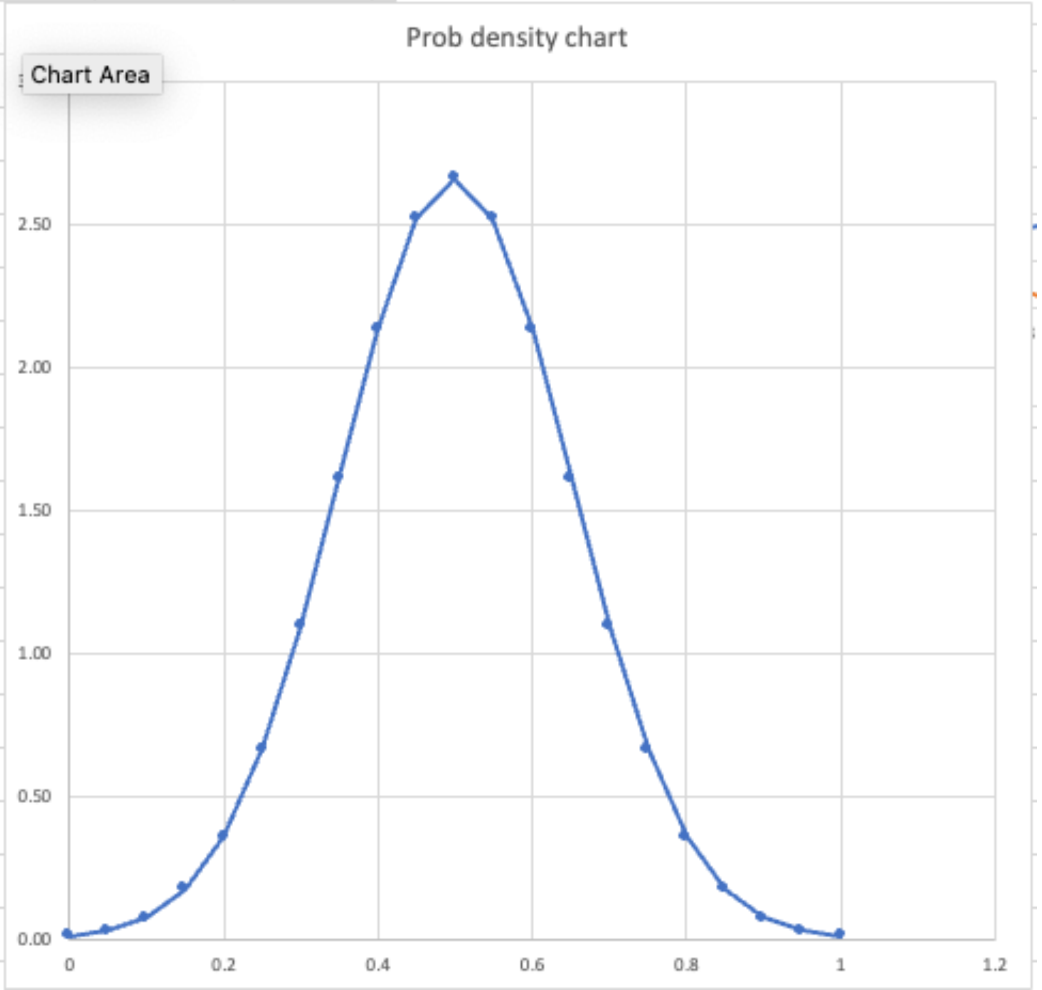
CPU Utilization

Mean = 50% utilization

STDev= 15%

Prob of CPU between

60% and 80% = 5.86%

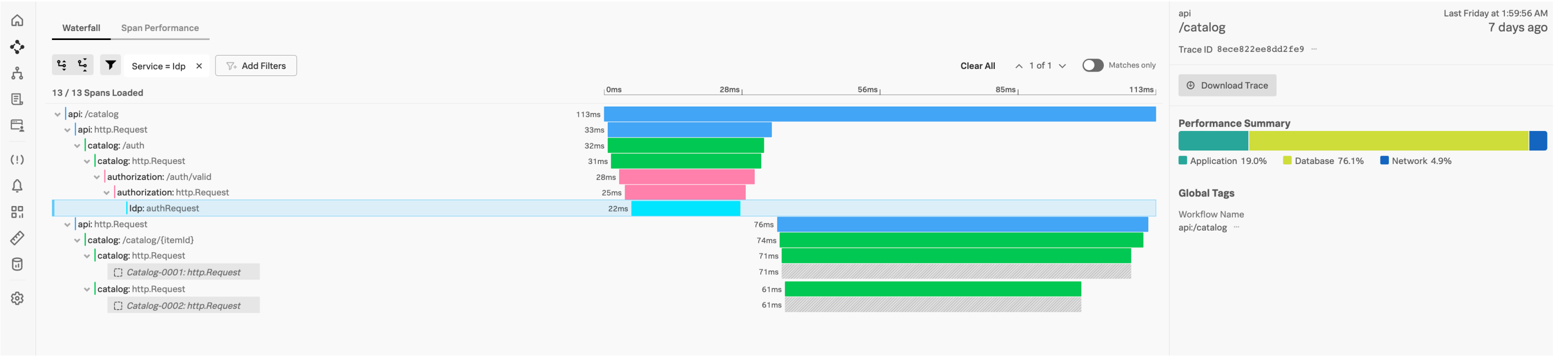


Slight sidetrack: Descriptive versus Inferential stats

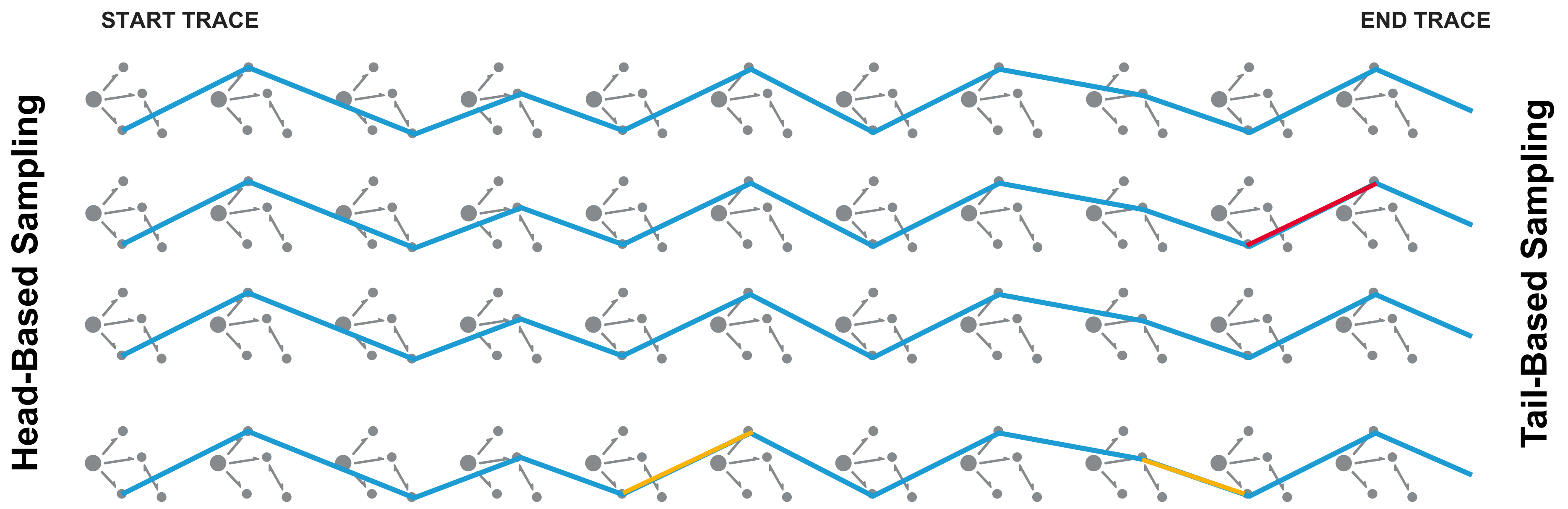
- Descriptive uses the whole data set to draw statistical conclusions
  - Used for visualization
  - Can define and extract trends
- Inferential uses a sampled set to draw conclusions
  - Used for predictions or hypotenuse testing
  - Can also visualize
- But this leads us to sampling

Dealing with the data

- Monitoring is now a data problem
    - Observability signals: Metrics, Traces, Logs
- Analysis is often
    - Aggregated or Analyzed in segments: Time-defined
    - Sampled and inferential
        - **Random** sampling
        - Stratified sampling
        - Cluster sampling
        - Systematic sampling
        - **Purposive** sampling



Lets consider tracing and sampling



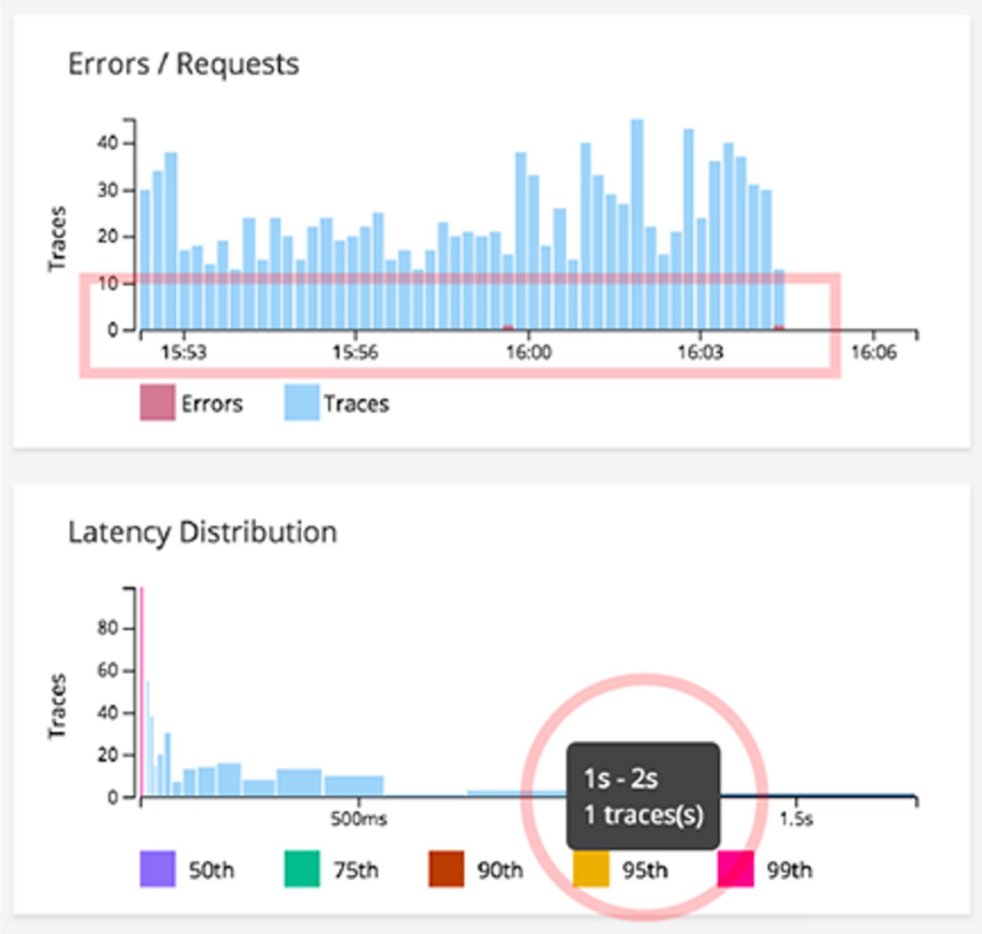
Purely random

Retention rate = Coverage

Not really random

Fault rate = Coverage

Sampling



No Sampling

Sampling

- Changes behavior from Descriptive to Inferential
- Can hide outlier behavior
  - Metrics are not usually sampled
- May make forensics tougher
  - Lack of direct correlation
- A necessary evil

Summary

- Statistics are how we tend to analyze our metrics
- Statistics are aggregation and reduction to reveal central tendencies
  - They do not show individual behavior
- Most choices make use of very few basics
  - But other choices may show amazing inferential results
- And finally

The most effective debugging tool is still careful thought, coupled with judiciously placed print statements.

-Brian Kernighan *Unix for Beginners* 1979

Thanks!

https://www.linkedin.com/in/davemc

NGINX Community Slack

