Know Your Data: The stats behind the alerts

Dave McAllister

Sr. OSS Technologist

NGINX

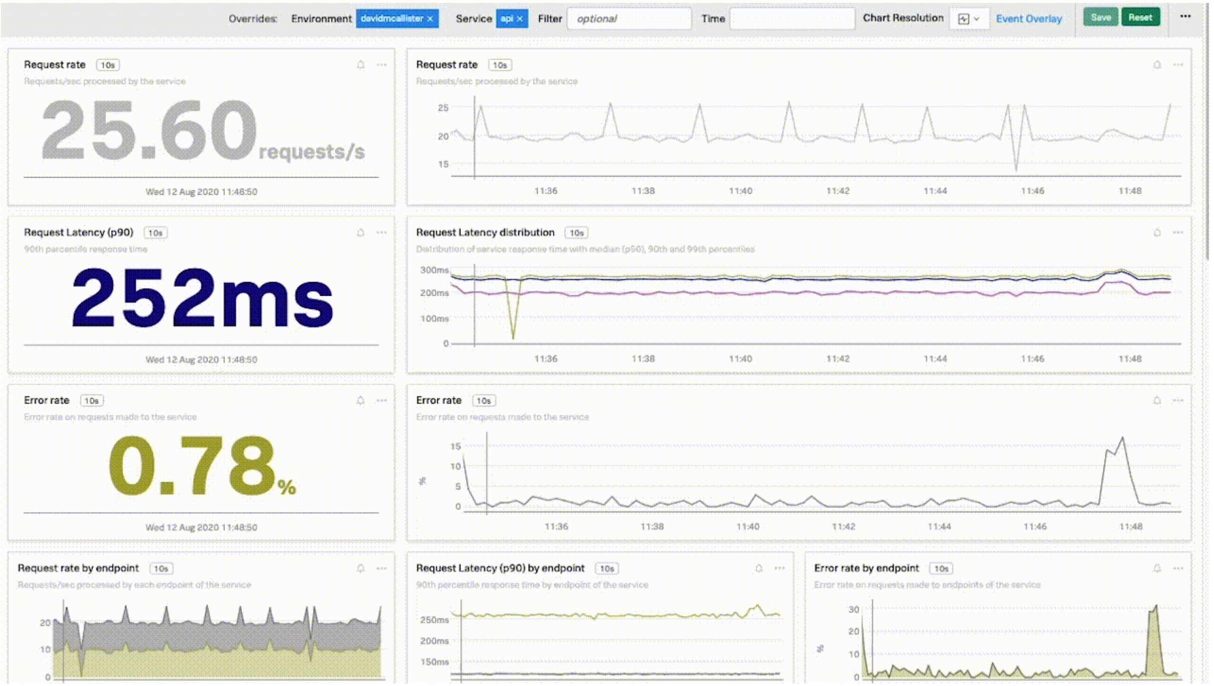
NGINX
Part of F5

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

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

And for extra credit, What’s the 9th Dedekind number?

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 when values build on each other?
- How do you arrive at values to represent rate of change over time?

Do you know what your alert is really showing you?

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:

Measure of central tendency, represents average value of a set of data

Mean = 5.444

Median:

Represents the middle value in a set of *ordered* data

Median = 6

Mode:

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:

Measure of central tendency, represents average value of a set of data

Mean = 5.444

Or is it 4.130 or 2.791?

Median:

Represents the middle value in a set of *ordered* data

Mode:

Value that appears most frequently in a set of data

Means to an End

Arithmetic, Harmonic, Geometric, Trimmed, Weighted, Moving

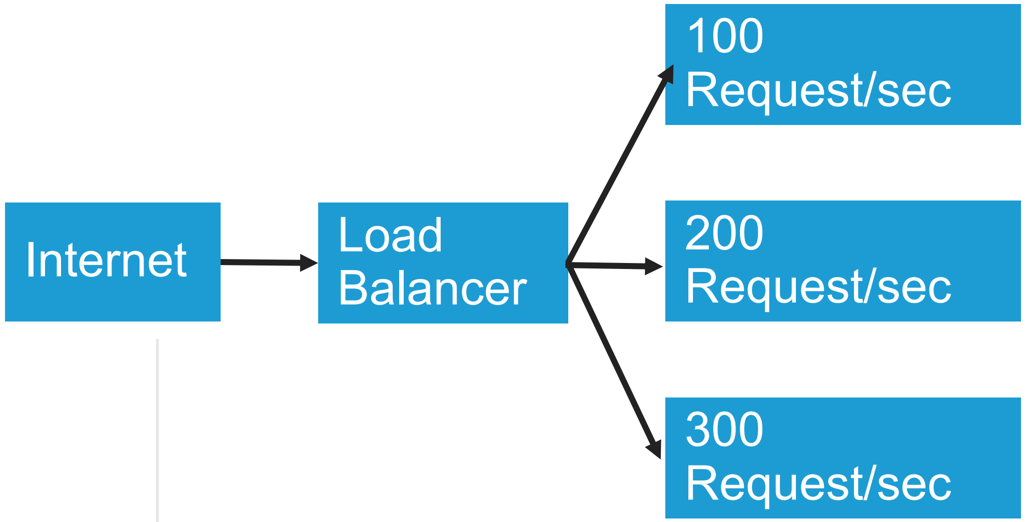
- Each has potential uses and drawbacks
- Often already implemented in monitoring software
- Can give very different results
- Can make like and unlike comparisons easier

Arithmetic

Also often called Average

- 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 = (100 + 200 + 300) / 3

Amean = 200 Requests per second

Geometric

- Multiply all the items together, take the nth root
- Often used for things growing exponentially

- In DevOps
  - Average number of deploys per unit of time
  - Average lead time for changes
  - MTTR
  - Throughput

DevOps team: optimizing app deployment

Sprint 1: 5% reduction

Sprint 2: 10% reduction

Sprint 3: 3% reduction

Sprint 4: 7% reduction

How well are they reducing deployment times?

X_1=1-0.05 = 0.95, x_2 = 0.90, X_3 = 0.97, x_4 = 0.93

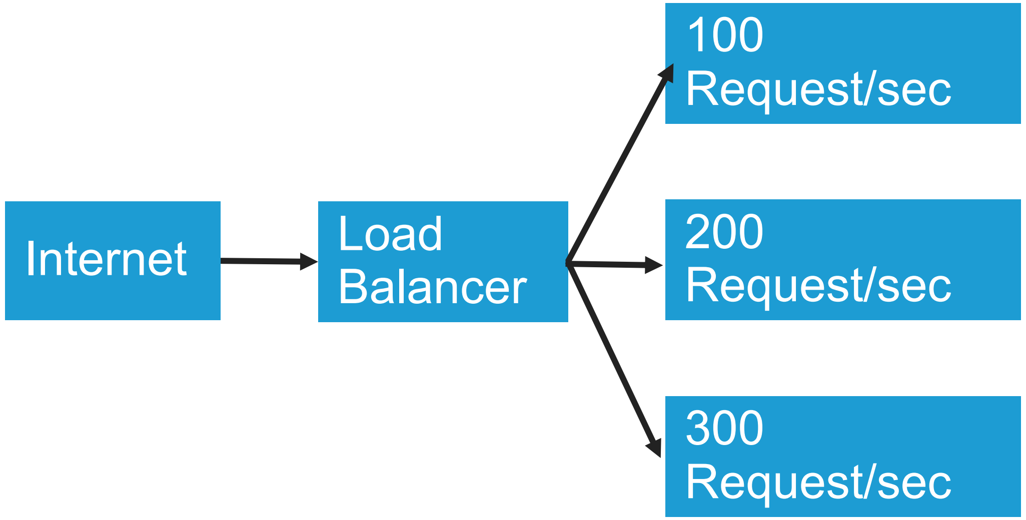
$$GM = \sqrt[4]{0.95 \times 0.90 \times 0.97 \times 0.93}$$

GM = 0.937 or

Deployment has a 6.3% improvement

Harmonic

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



$$HM = \frac{3}{\frac{1}{100} + \frac{1}{200} + \frac{1}{300}}$$

HM = 150 Requests per second

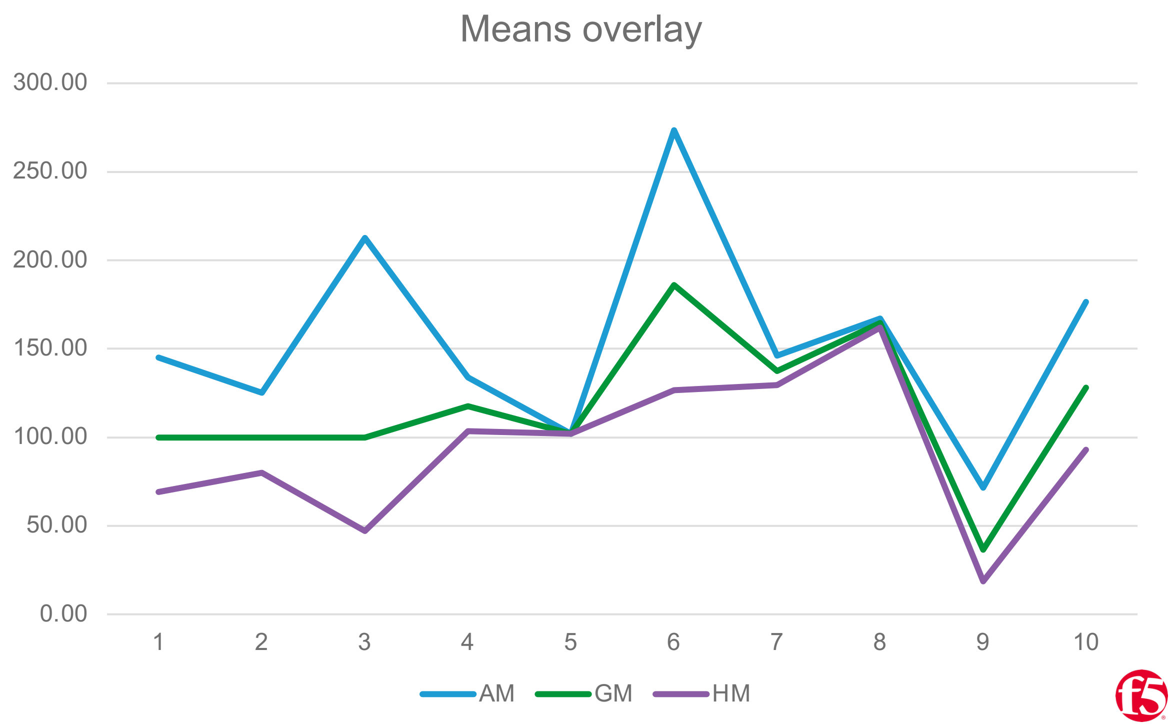
Take a comparison look

Average Latency ms/sec	Average Count/sec	Suggested metric	AM	GM	HM
250	40	10000	145.00	100.00	68.97
200	50	10000	125.00	100.00	80.00
400	25	10000	212.50	100.00	47.06
198	70	13860	134.00	117.73	103.43
105	99	10395	102.00	101.96	101.91
474	73	34602	273.50	186.02	126.52
195	97	18915	146.00	137.53	129.55
196	138	27048	167.00	164.46	161.96
133	10	1330	71.50	36.47	18.60
298	55	16390	176.50	128.02	92.86

What's more important to the equation?

- Latency
- Thruput

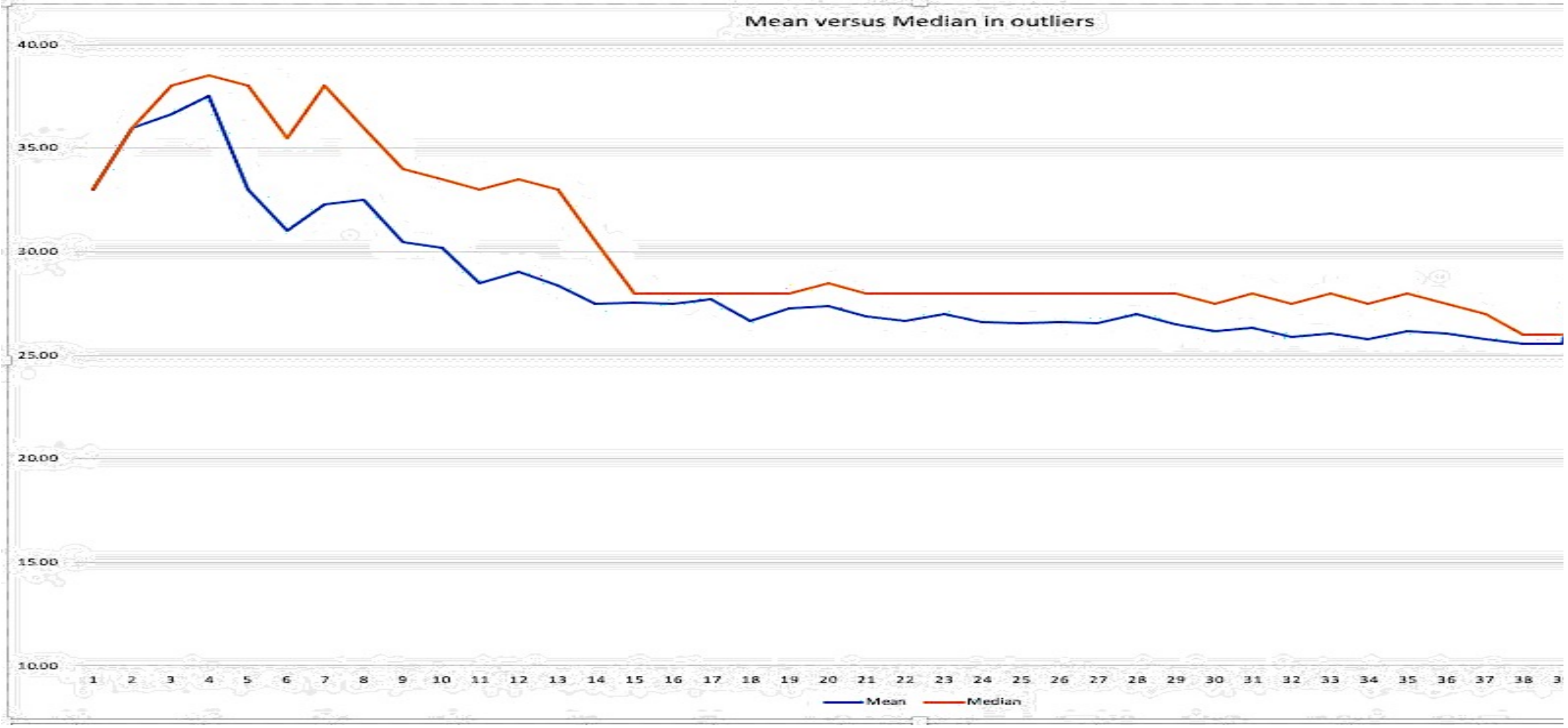
- GM is the moral equivalent
- AM weights the larger number
- HM weights the lower value



Median

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

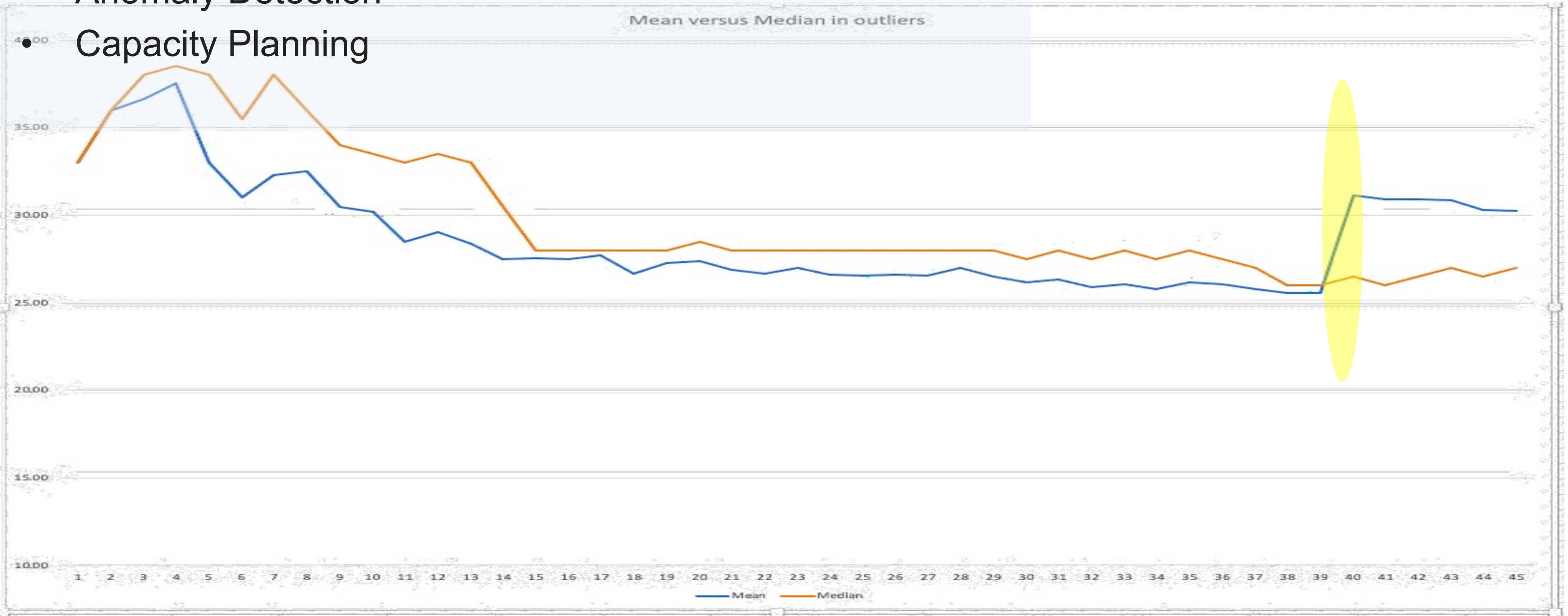
Mean	25.25
Median	26

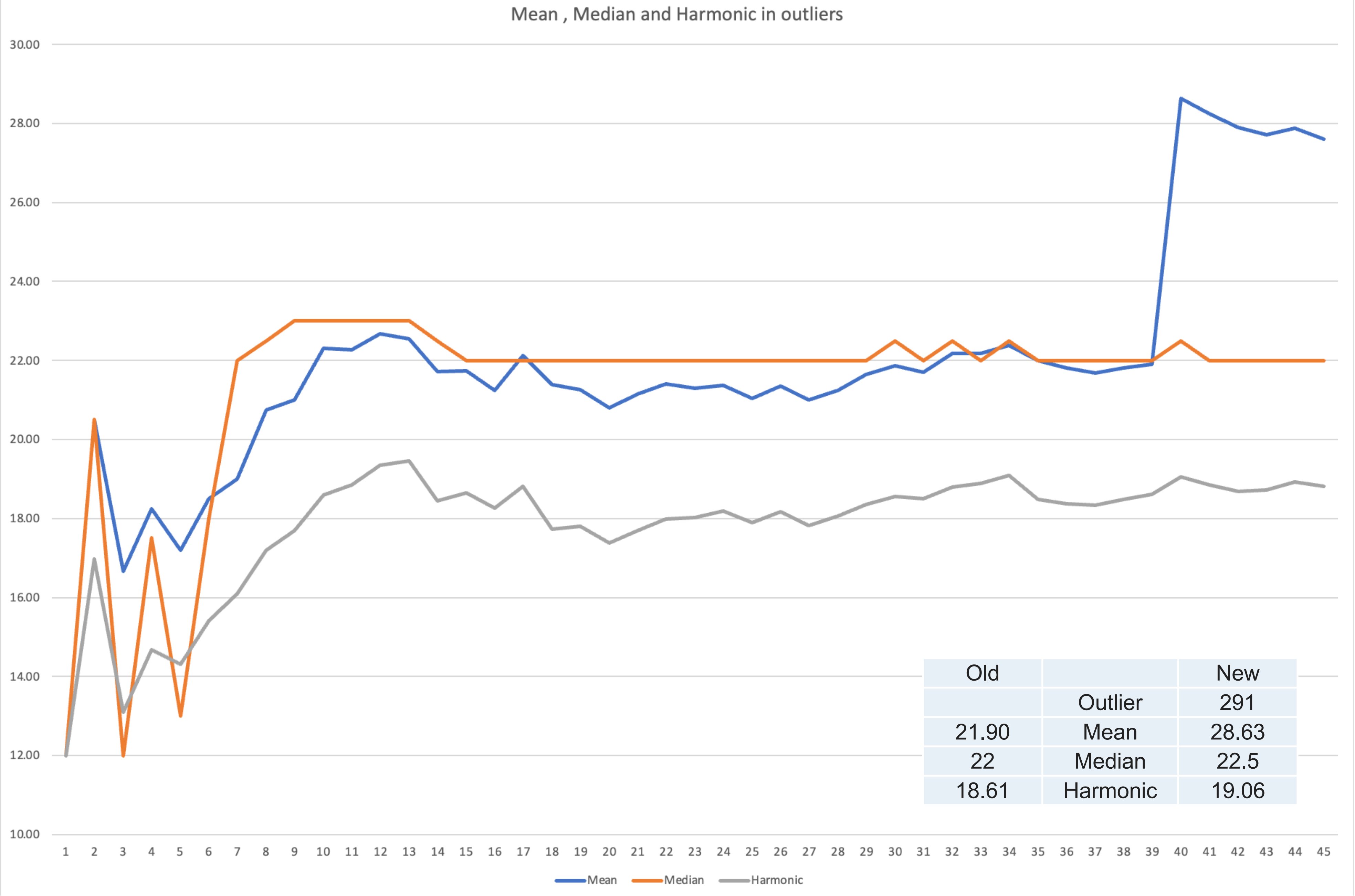


Choosing Between Mean and Median

- 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

If you are using P95

you are using using a percentage value

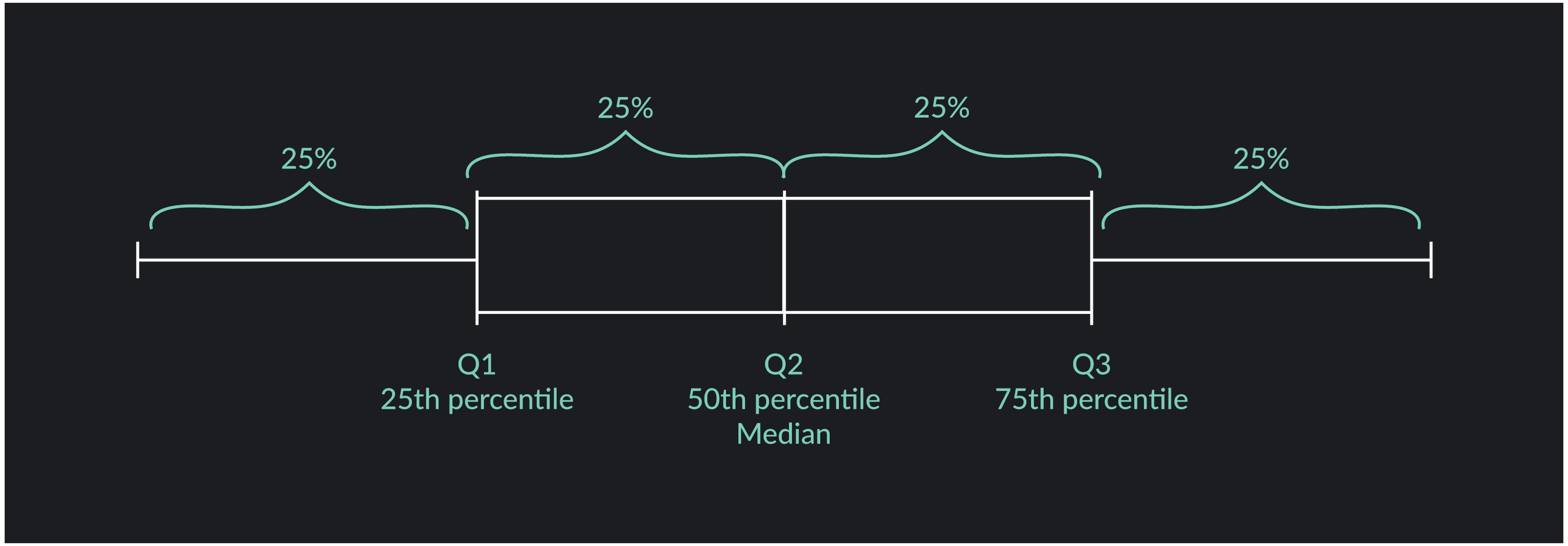
Congrats!

Slight sidetrack: Measure of Variability

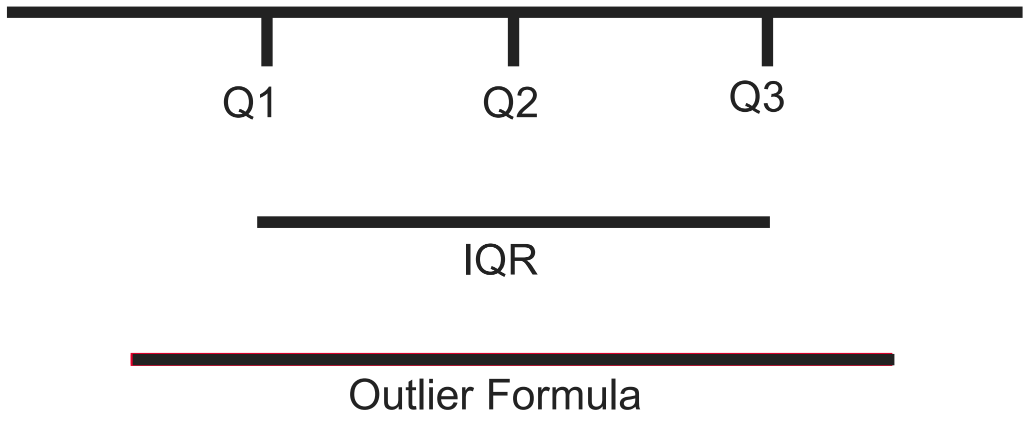
How the numbers behave.

- Standard Deviation
- Range
- InterQuartile Range (IQR)
- Variance
- Clusters
- Outliers

Properly used, variability can help you target outliers



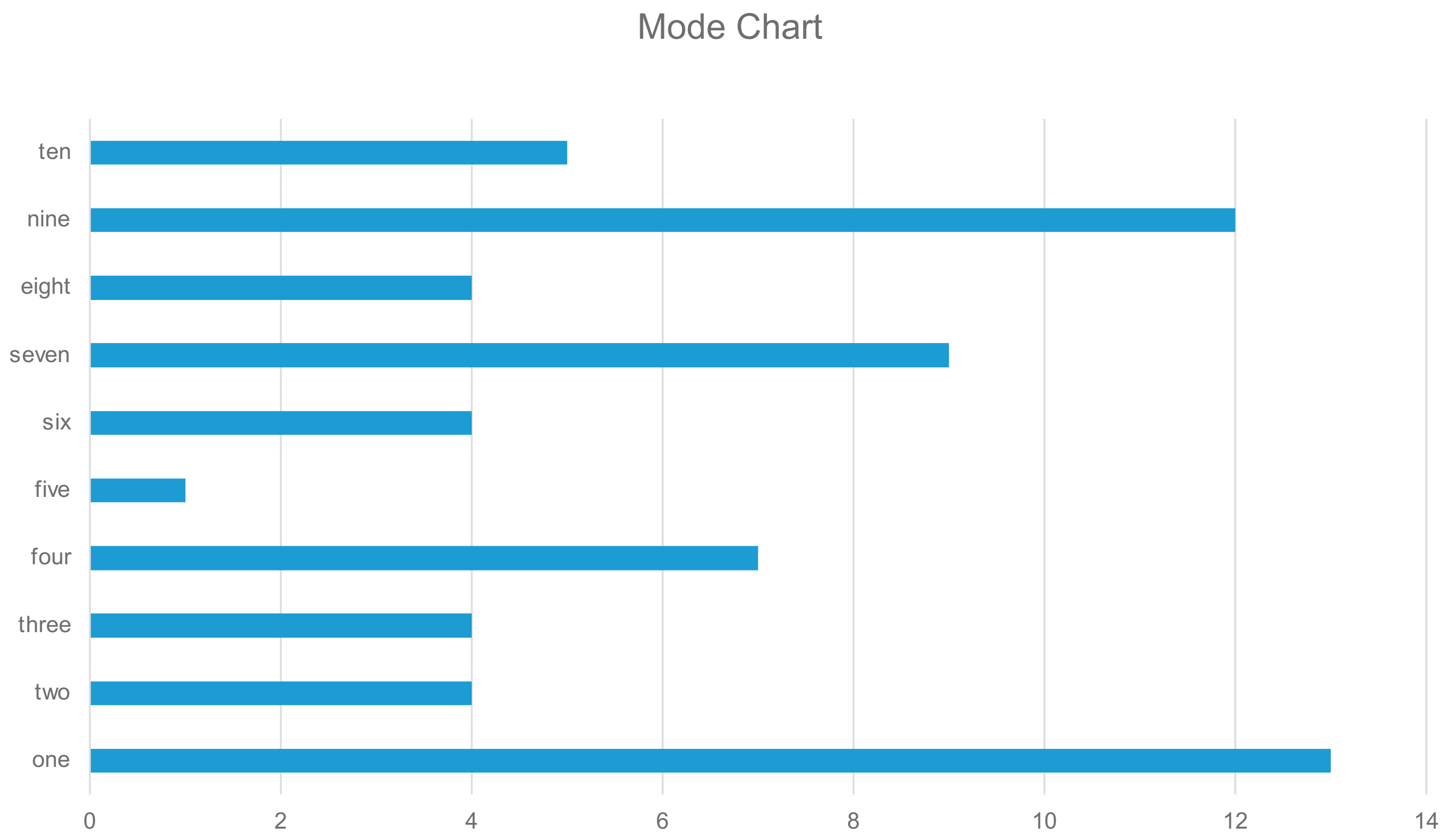
Outlier Formula, visually



Those outside edges are good candidates for outliers

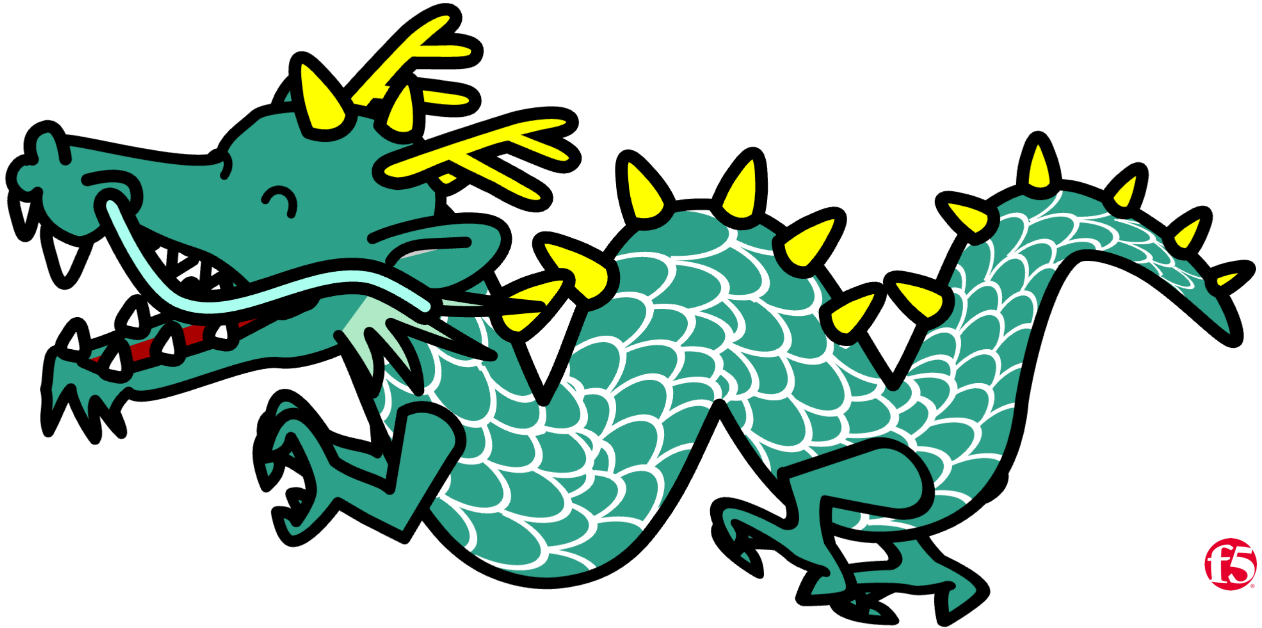
How about the Mode?

- 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



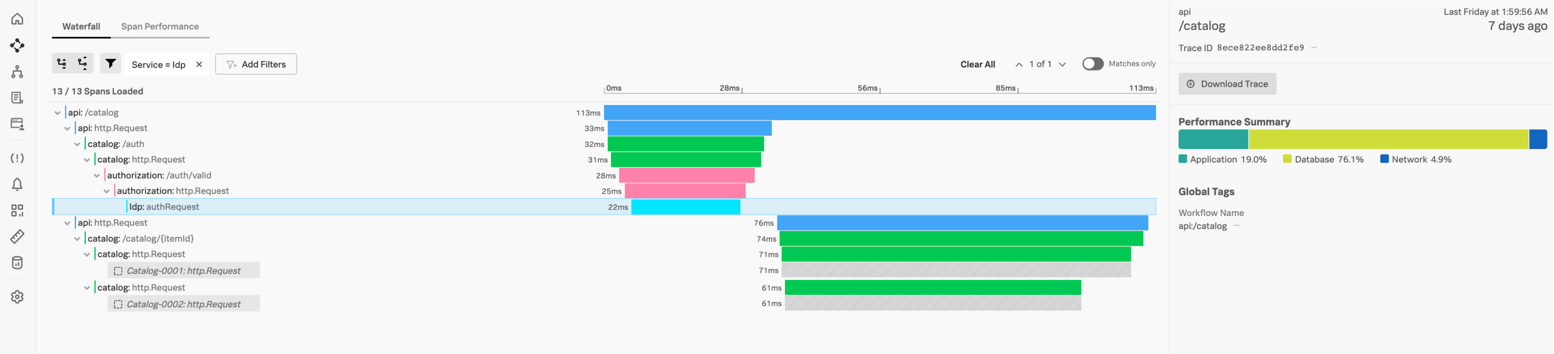
Slight sidetrack: Descriptive versus Inferential stats

- 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

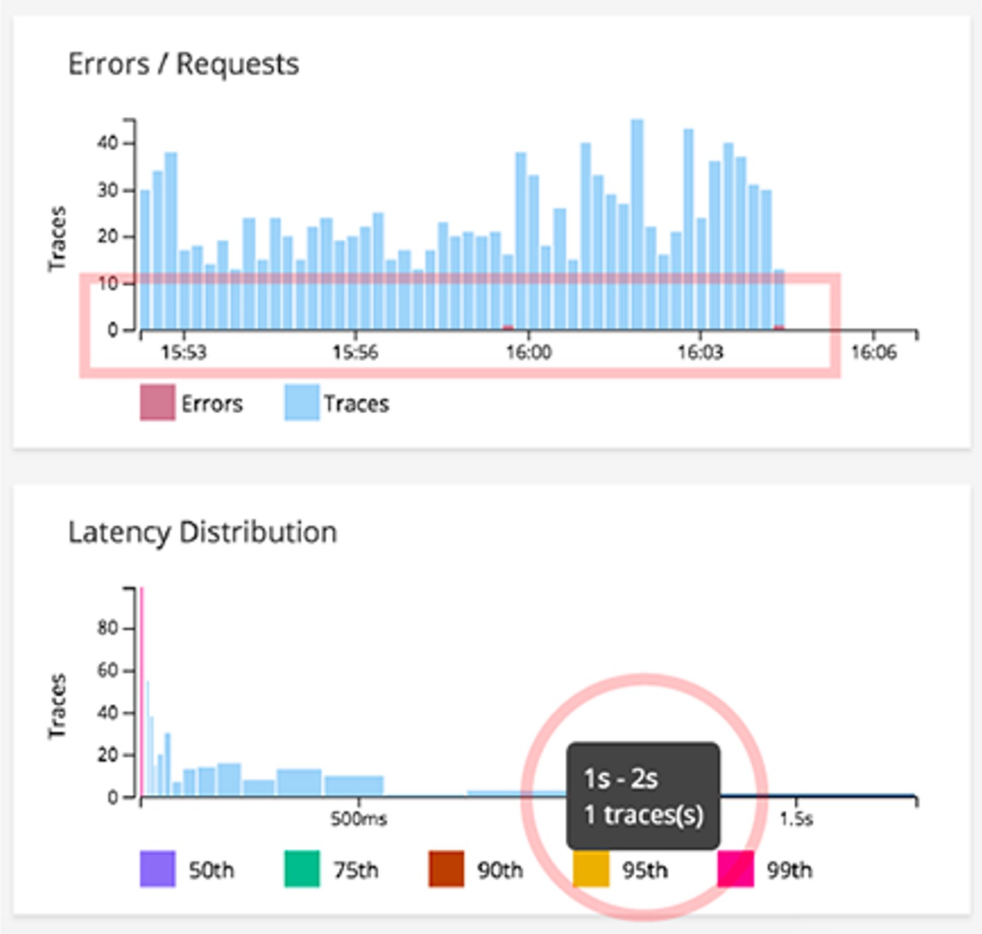


Dealing with the data

- 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



Sampling



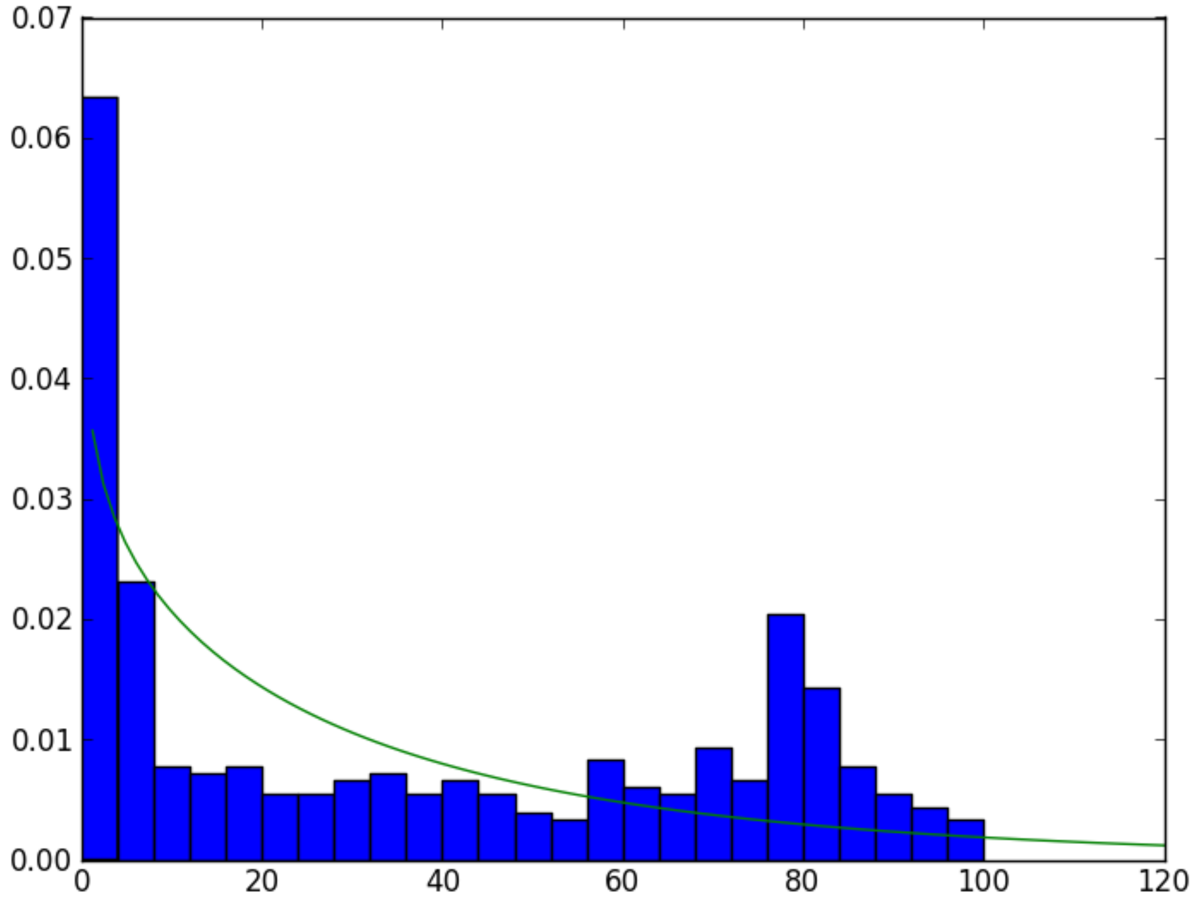
No Sampling

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
  - But understand what you are giving up

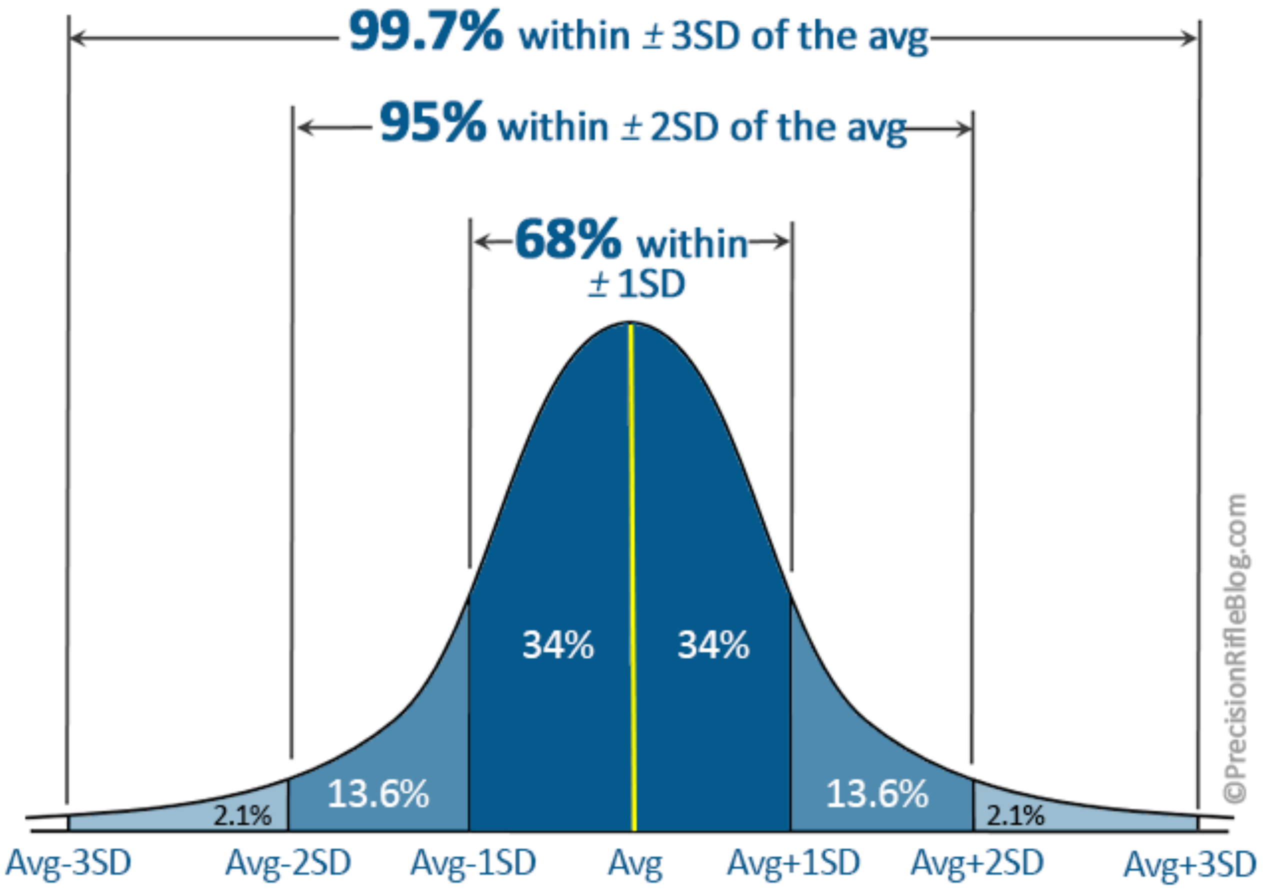
Distributions

- 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
    - $F(t)=1-e-(t/\lambda)\beta$
    - 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 : shape 1.0624082 scale 2158.256

$Fdisk(300)\approx1-e-0.6918\approx0.50025$
$Fmemory(300)\approx1-?-0.3490\approx0.2959Fmemory(300)\approx1-e-0.3490\approx0.2959$
$Fpower(300)\approx1-?-0.2327\approx0.2092Fpower(300)\approx1-e-0.2327\approx0.2092$
$FCPU(300)\approx1-?-0.1745\approx0.1593FCPU(300)\approx1-e-0.1745\approx0.1593$
$FSSD(300)\approx1-?-0.1396\approx0.1298FSSD(300)\approx1-e-0.1396\approx0.1298$

```
p.failure <- pweibull(300, shape = fit.weib$estimate[1],
scale = fit.weib$estimate[2])
1 - p.failure
```

Failure (disk, memory) = 64.81%
Failure (entire system) = 75.52%

Deeper dive: Exponential

- Models the "rate" (time between events that are unrelated)
- Use cases
  - Network performance
  - User Requests (traces)
  - Messaging service
  - System failures
- $f(x) = me^{-mx}$
  - λ = 1/μ
  - e ~ 2.71828182846

Average Latency ms/sec	Average Count/sec	Suggested metric	AM	GM	HM
250	40	10000	145.00	100.00	68.97
200	50	10000	125.00	100.00	80.00
400	25	10000	212.50	100.00	47.06
198	70	13860	134.00	117.73	103.43
105	99	10395	102.00	101.96	101.91
474	73	34602	273.50	186.02	126.52
195	97	18915	146.00	137.53	129.55
196	138	27048	167.00	164.46	161.96
133	10	1330	71.50	36.47	18.60
298	55	16390	176.50	128.02	92.86

<u>Latency</u>

λ = 1/244.9

Probability density of 158 ms = 0.2142%

Cumulative density = 47.5422%

Expdist median ≈ 169.7 ms

<u>Throughput</u>

λ = 1/65.7

Probability density at 58 count = 0.628%

Cumulative density = 58.7%

Expdist median ≈ 45.6

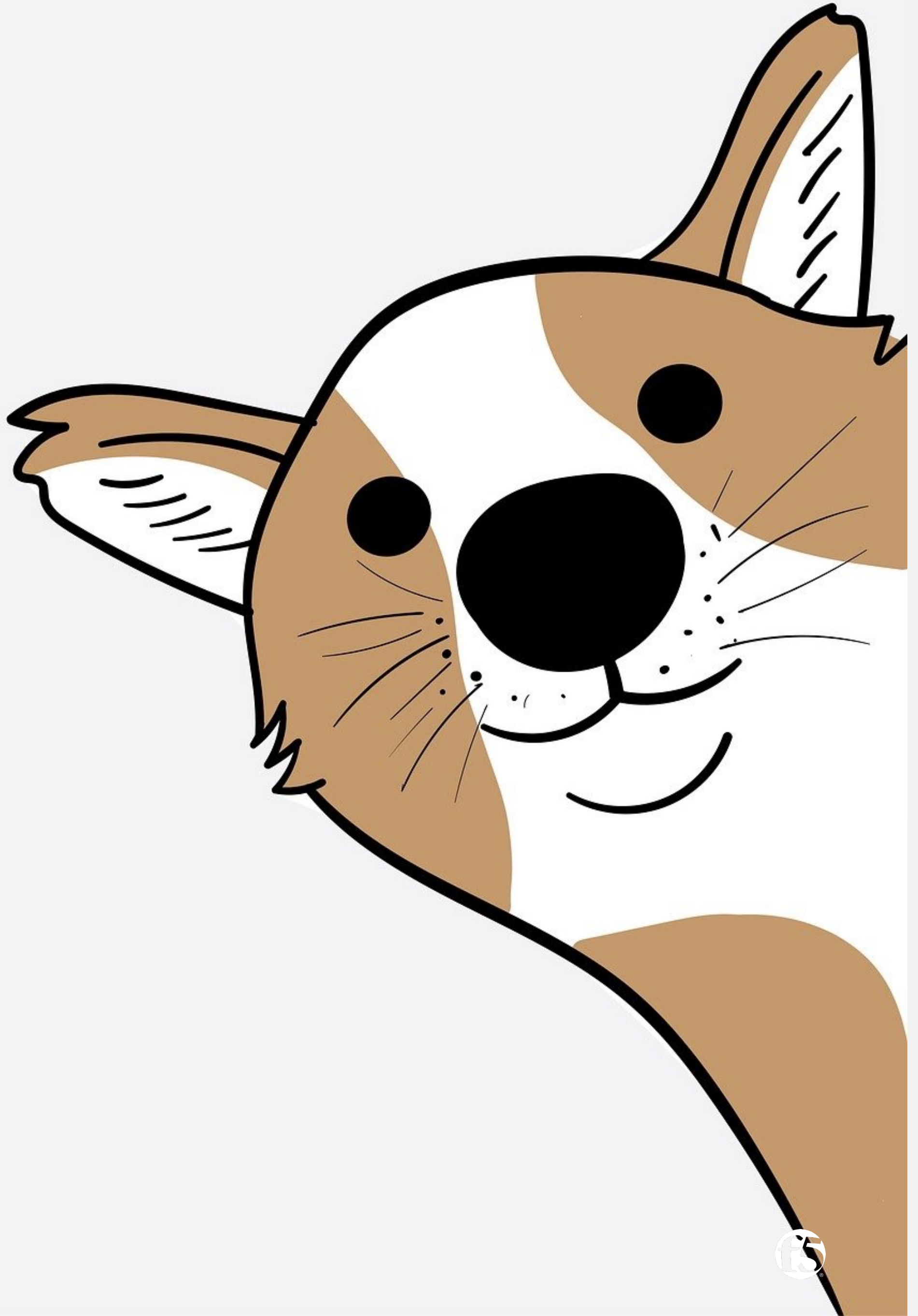
Slight Sidetrack and a pet peeve

You may stumble upon:

“On scale, *statistics* are not your friend”

WRONG

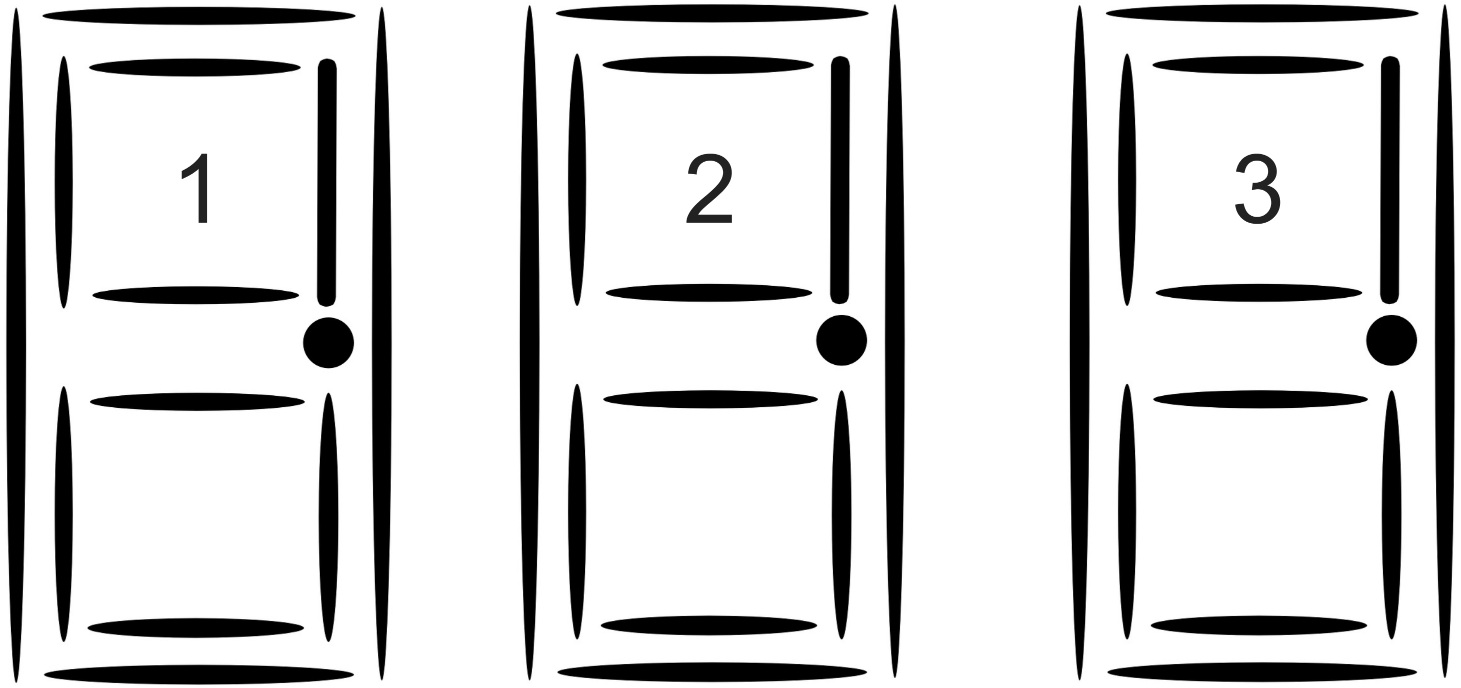
On scale, *probability* is not your friend.



Bayes Theorem

Conditional Probability – The Monty Hall Problem

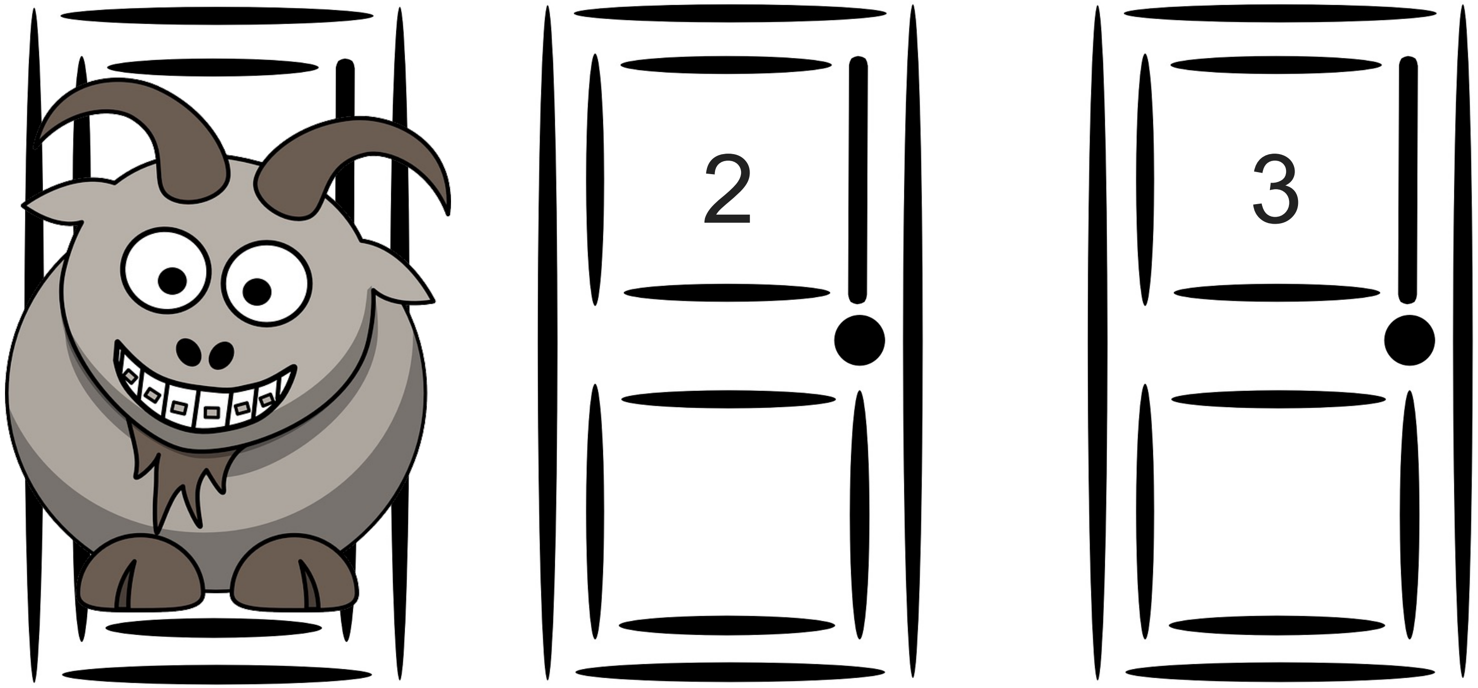
- You have three doors
- Behind one of them is a car
- The other two are goats
- You pick Door # 2



Bayes Theorem

Conditional Probability – The Monty Hall Problem

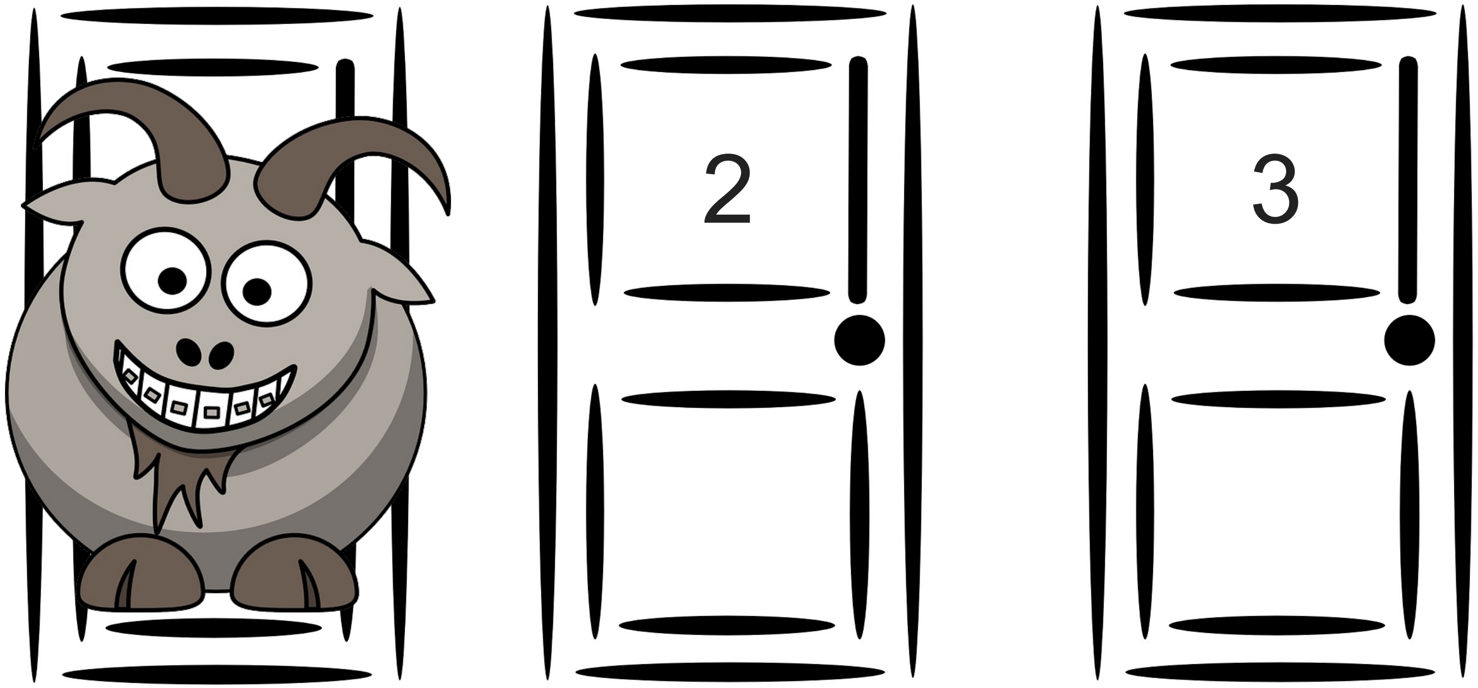
- You have three doors
- Behind one of them is a car
- The other two are goats
- You pick Door # 2
- The host shows you door # 1 and offers to let you switch
- Do you switch or not?



Bayes Theorem

Conditional Probability

- Mathematical framework for updating probability for an event as new information becomes available
- Based on this theorem
  - Initial probability of Door 2 = 1/3
  - Initial probability of Doors 1 or 3 = 2/3
- Since it isn't behind Door # 1 then Door # 3 is the remaining 2/3



Bayes Theorem in DevOps

Predictive Monitoring

- Historically there's a 5% chance that your application will crash on any given day. Now, you get alerts about increasing memory consumption.
- Bayes' theorem indicates the probability of an impending application crash.

Incident Troubleshooting

- You're seeing an increase in error rates after a recent deployment. Historically, 80% of such incidents have been tied to database issues after major updates.
- Bayes' theorem suggests that the current issue is database-related considering the recent deployment and historical data.

Optimizing A/B Testing

- You're testing a new feature against a baseline to determine its impact on server load. Initially, both variants might be assumed equally likely to have an effect.
- Data from the test allows you to update the probability distribution of the effect of the new feature on server load.

Bayes' theorem provides a mathematical framework for incorporating new evidence into existing beliefs

Common Pitfalls in Statistics

Ignoring Scale

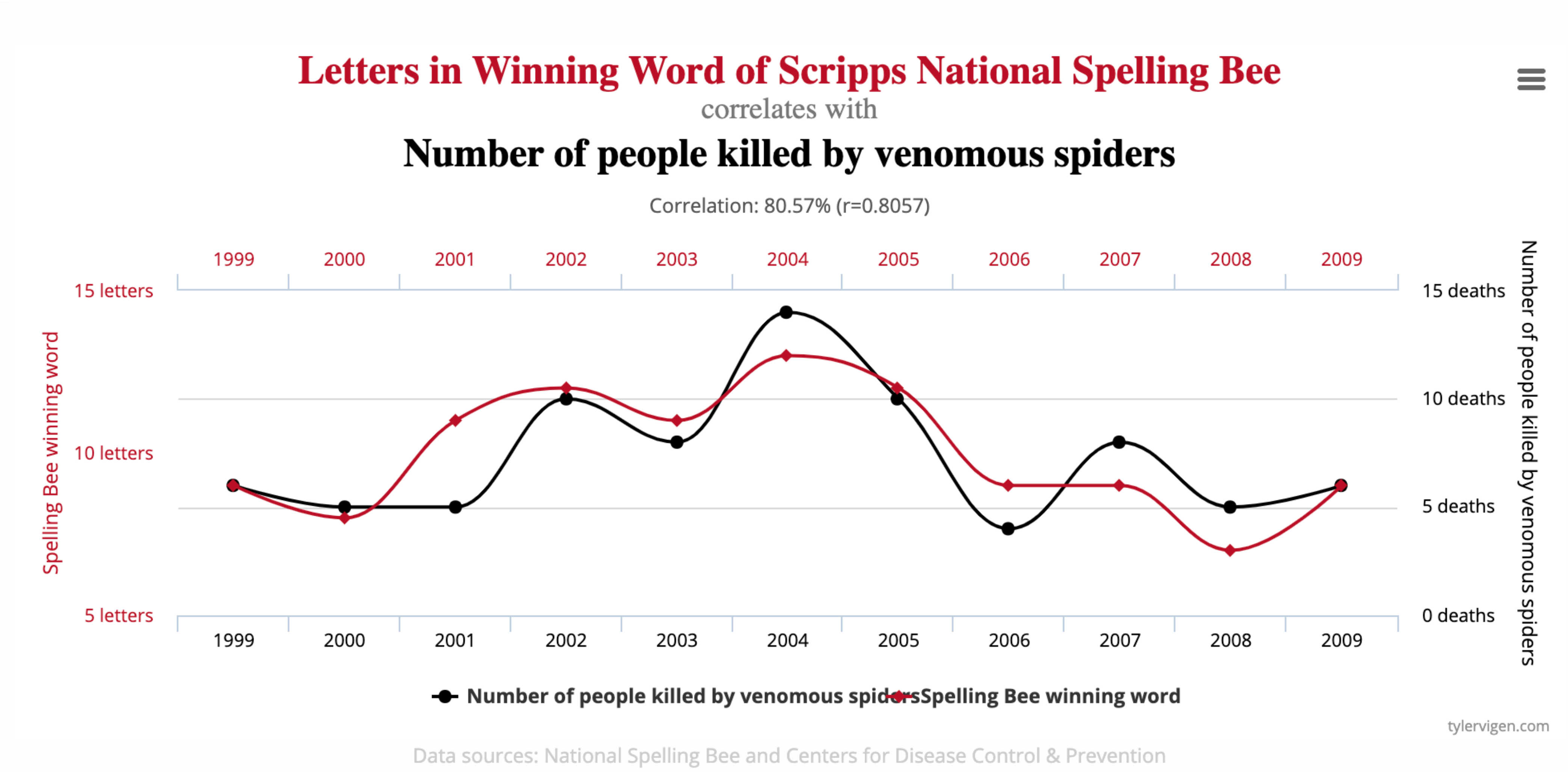
Looking at the wrong central measure

Confusing correlation with causation

Failing to see biases

Getting causation backwards

Correlation and Causation



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

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

D(9) = 286 386 577 668 298 411 128 469 151 667 598 498 812 366

Thanks!

Slides on GitHub

Linkedin: in/davemc

NGINX Community Slack

