Statistics for Engineers

SRECon EMEA 2022

Heinrich Hartmann | zalando.de | @HeinrichHartmann

Hi, I'm Heinrich

Bio

Head of SRE @ zalando

Data Scientist @ CIRCONUS

Mathematician @ The University of Bonn
Doctor of Philosophy (PhD), Matchematik
2008 - 2011

Talks & Publications

- Statistics for Engineers (SRECon 2015..2019)
- How to measure Latency (P99 Conf 21)
- State of the Histogram (SLOConf 2021)
- Latency SLOs Done Right (FOSDEM 2019)
- Circllhist - A Histogram Data Structure (arxiv)
- blog: heinrichhartmann.com
- twitter: @HenrichHartmann

Statistics for Engineers

Visualisations

Summary Statistics

Latency SLOs

Sampling

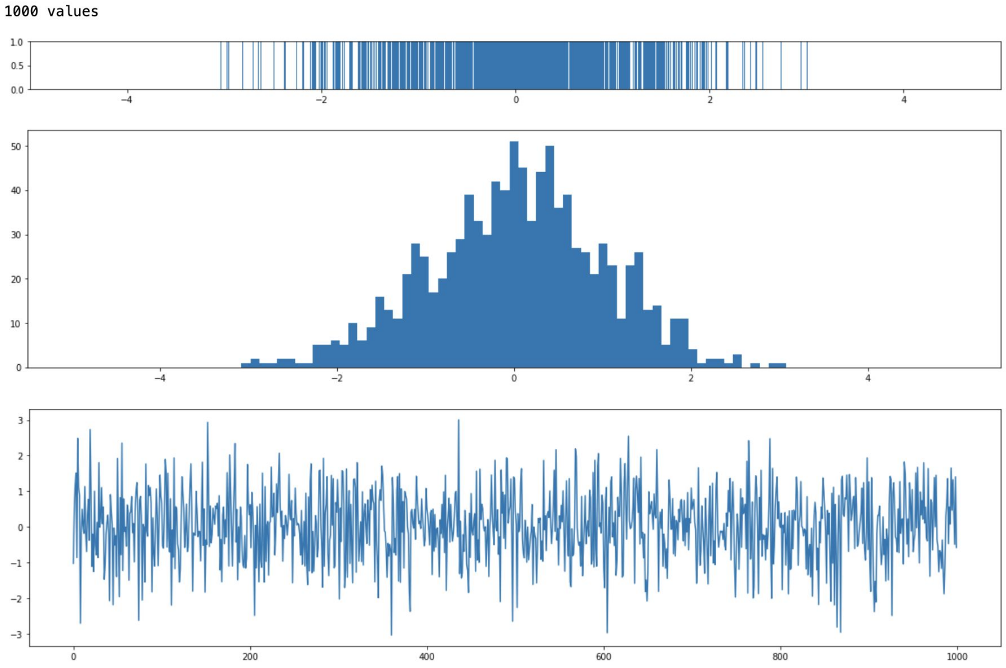
Visualizations

Example Dataset - Normal Noise

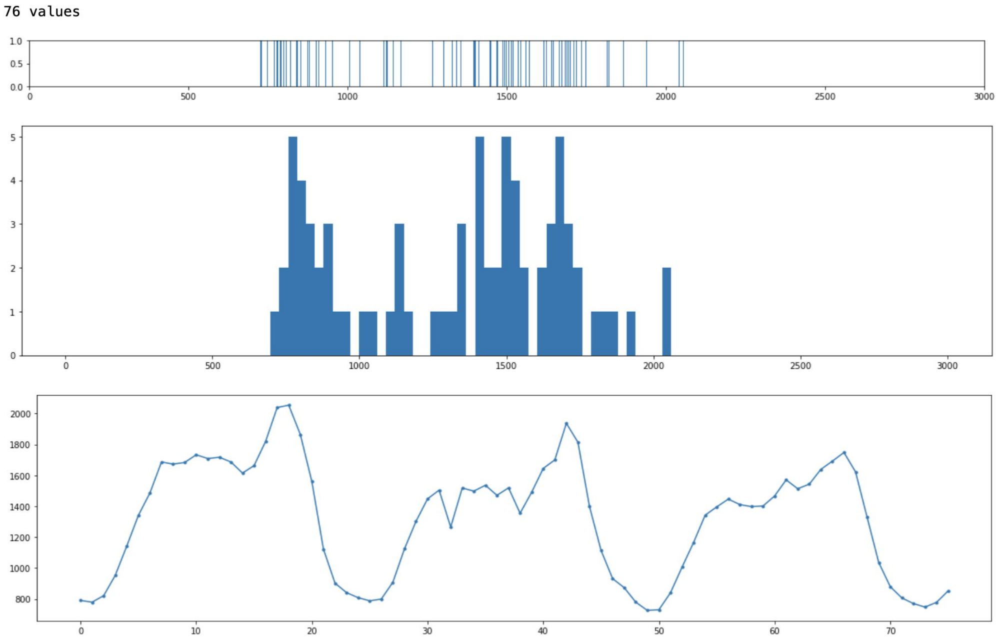
Rugplot

Histogram

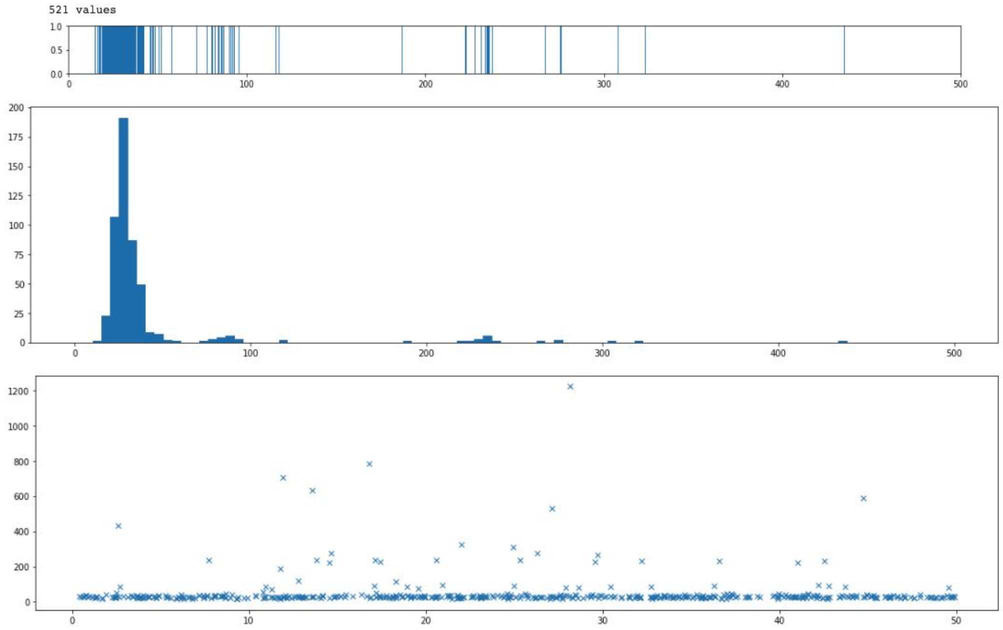
Linechart



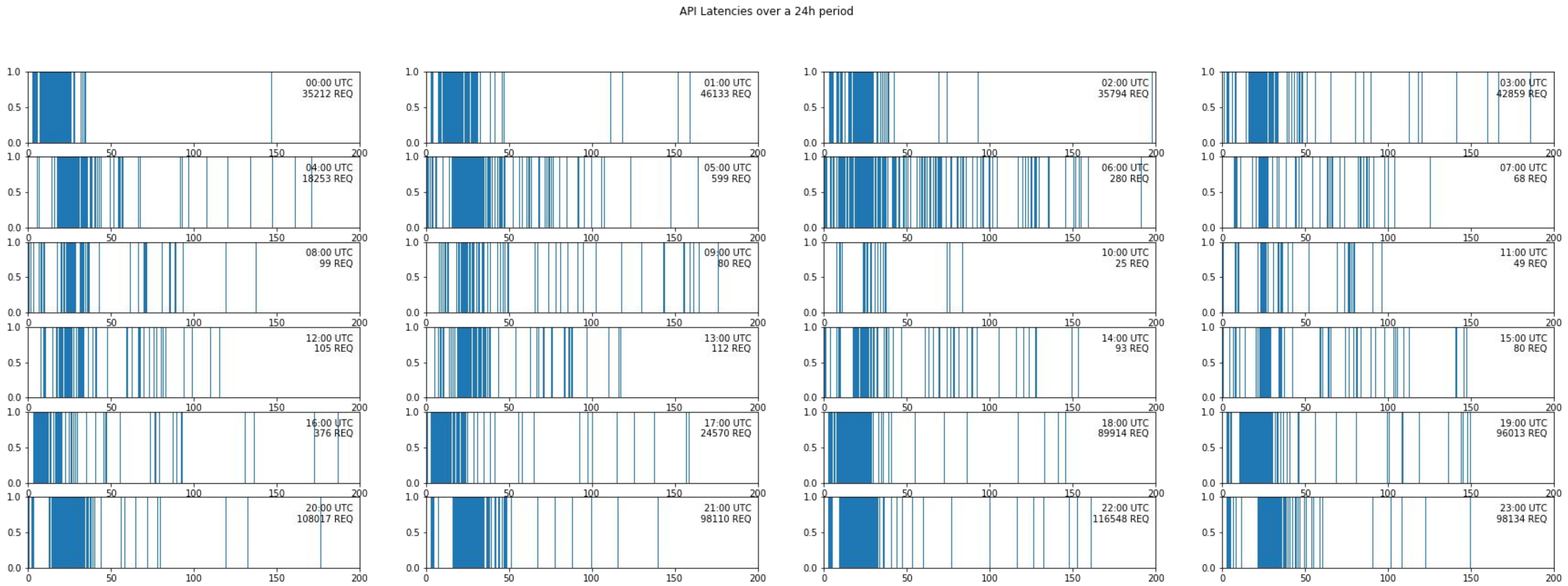
Example Dataset - Request Rates



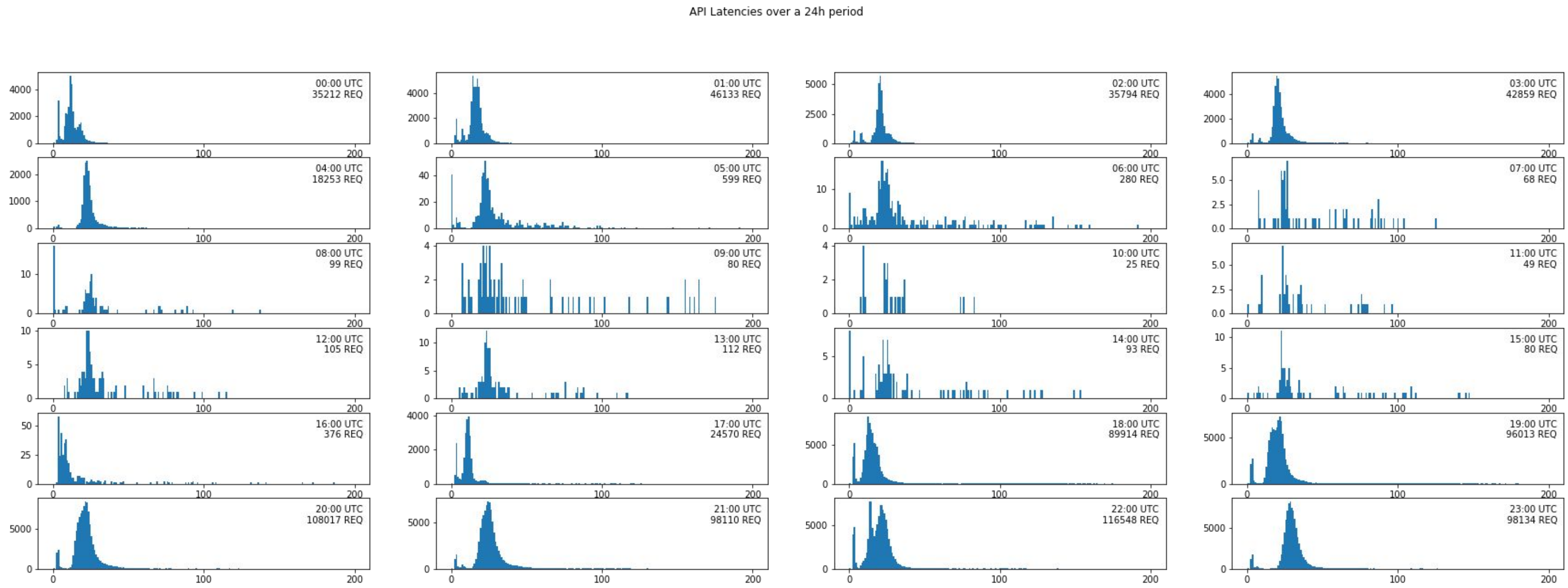
Example Dataset - Request Latencies



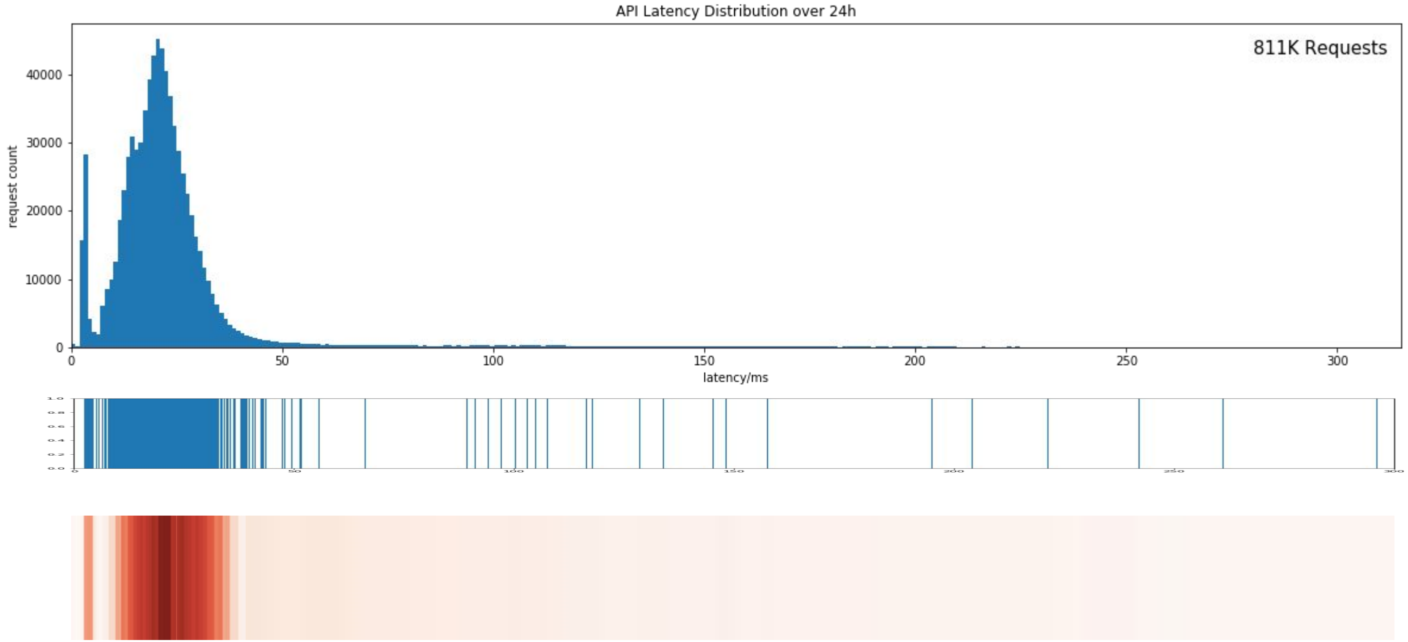
Example Dataset - Request Latencies over Time



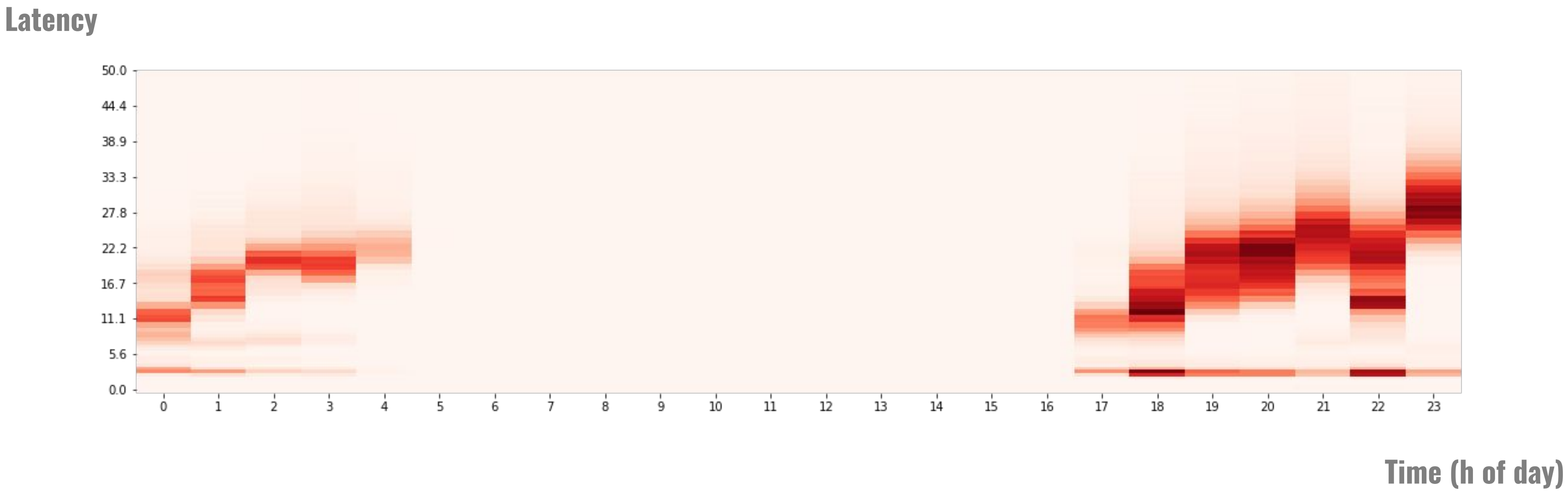
Example Dataset - Request Latencies over Time



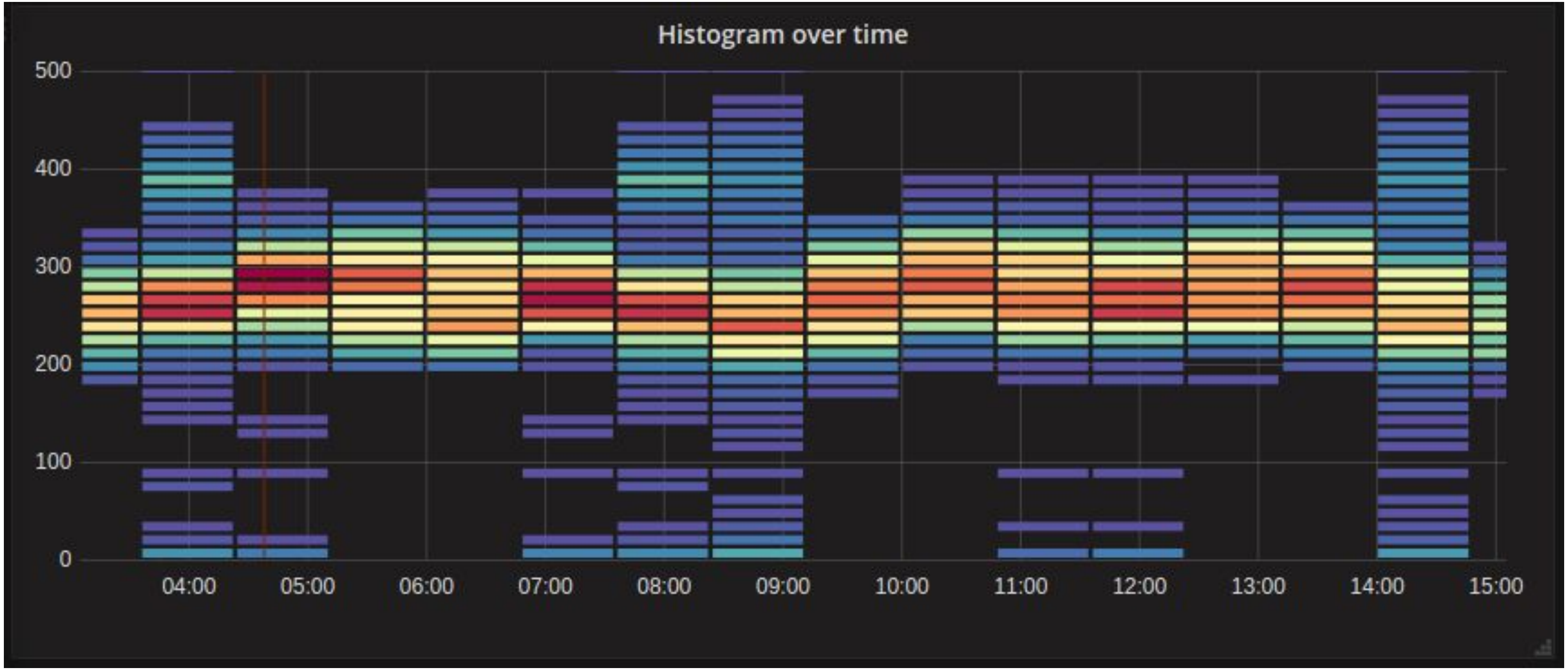
Density Heatmap



Example Dataset - Request Latencies over Time



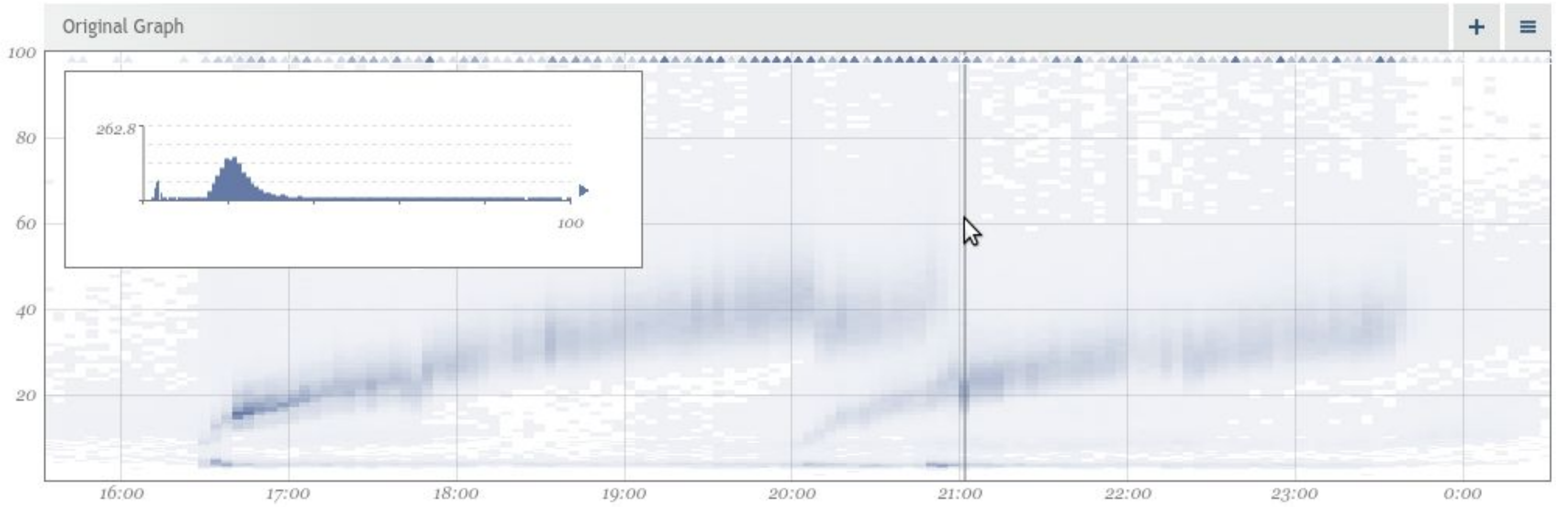
Grafana Histograms



source: https://grafana.com/docs/grafana/latest/fundamentals/intro-histograms/

Grafana

Production Example - Request Latencies over Time



CIRCONUS

Summary Statistics

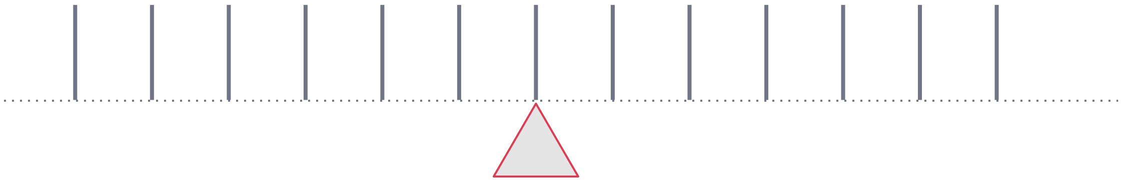
Aggregating Telemetry Data

- Across Time (Graphs, SLOs)
- Across Hosts / Containers (www-*)
- Across Endpoints (/, /posts, /archive)

AVERAGES

Summary Statistic - Averages

$$\mu = mean(X) = \frac{1}{n}\sum_{i=1}^{n} x_i.$$



Summary Statistic - Averages

$$\mu = mean(X) = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

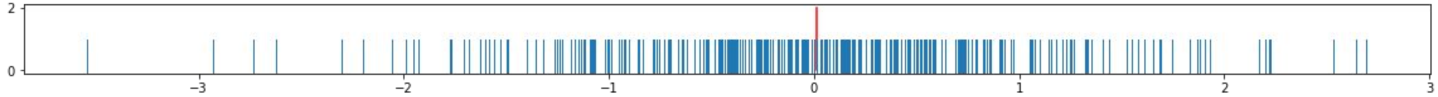
Summary Statistic - Averages

$$\mu = mean(X) = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

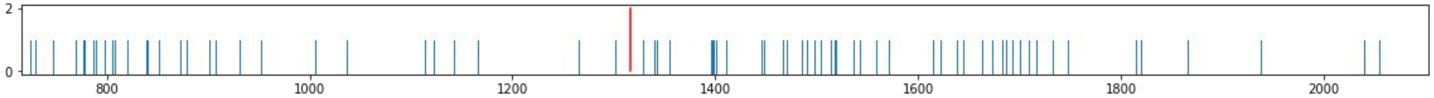
Summary Statistic - Averages

$$\mu = mean(X) = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

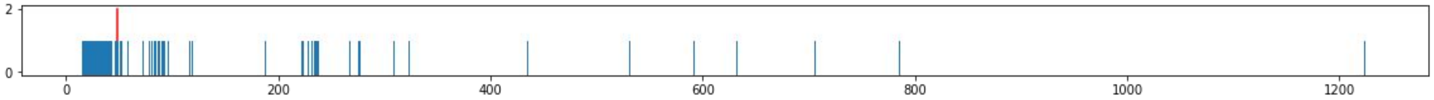
Normal Noise



Request Rates



Request Latency

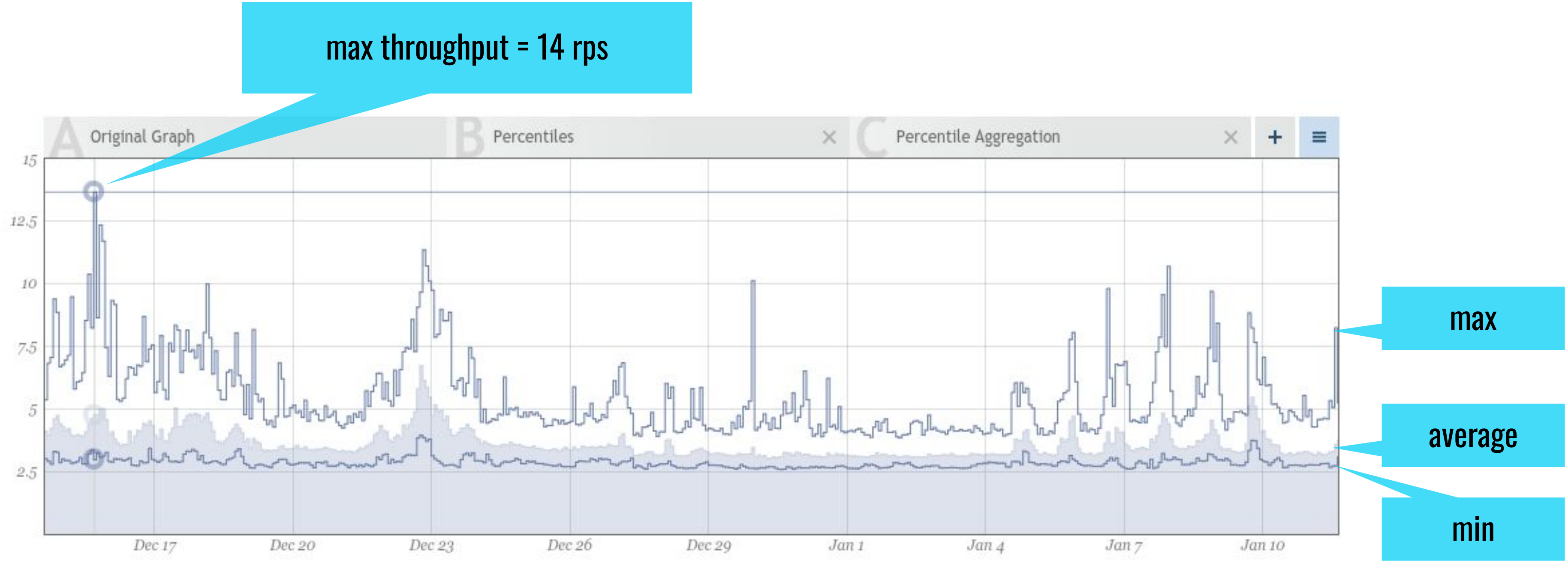


Can you spot the average?

CIRCONUS

Spike Erosion - Full Picture

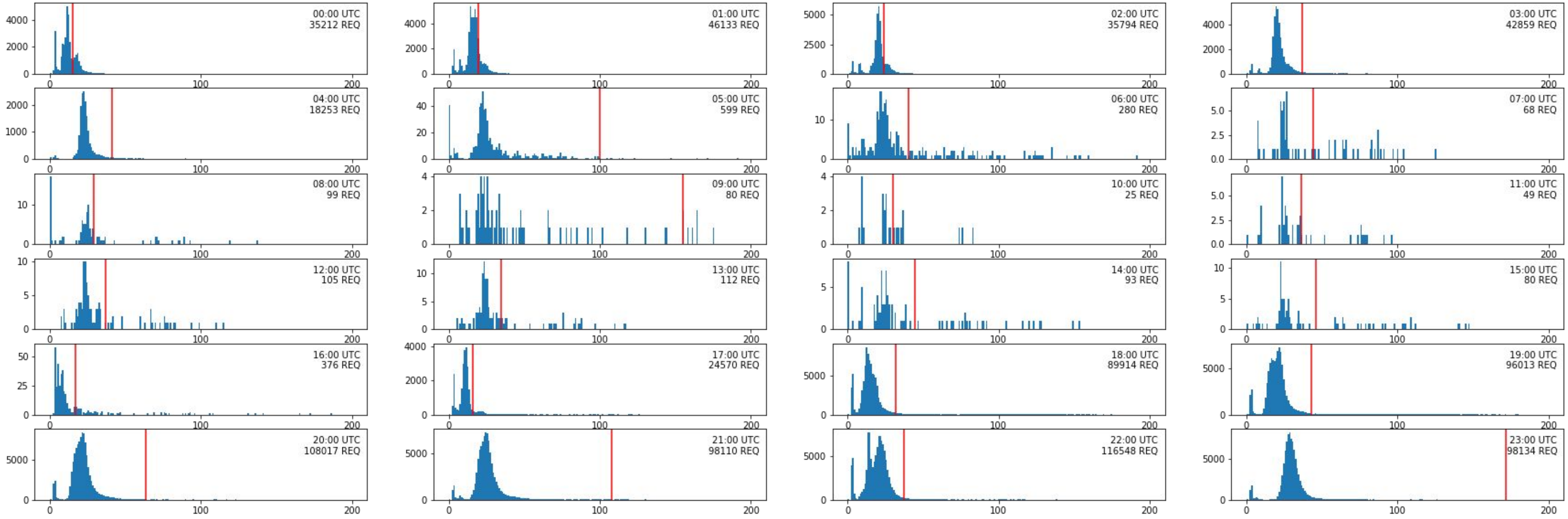
more on blog.circonus.com - Show me the Data (2016)



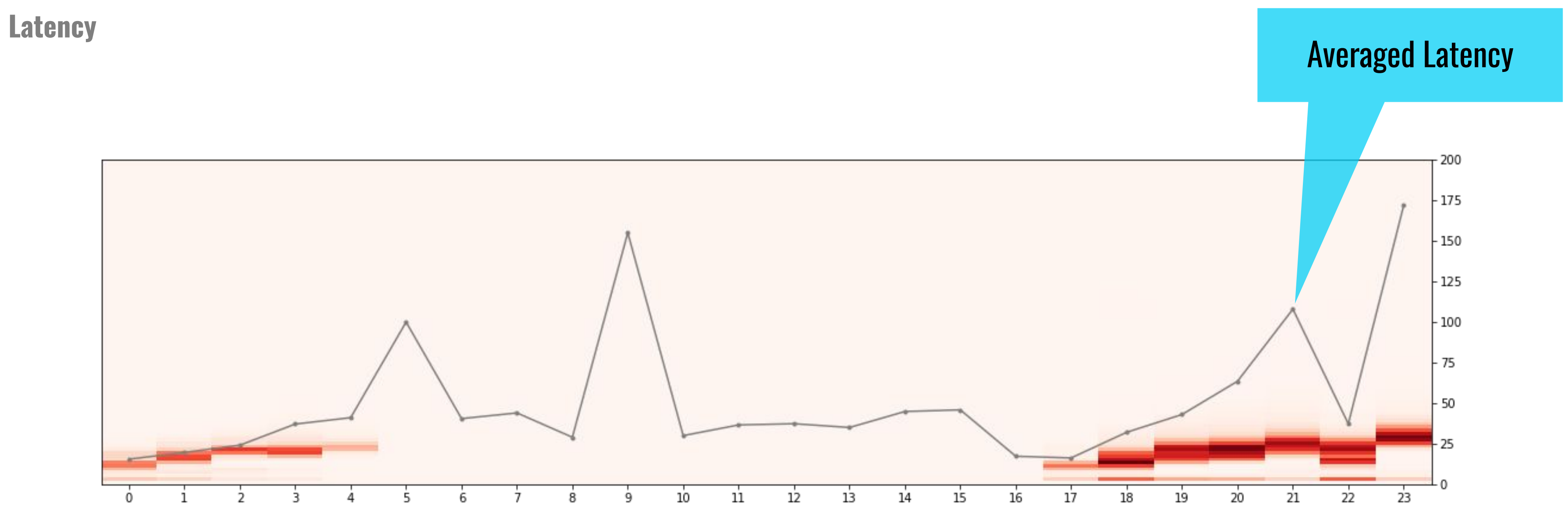
Example Dataset - Request Latencies over Time

$$\mu = mean(X) = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

API Latencies over a 24h period



Example Dataset - Request Latencies Mean Values



“

"Looking at your average latency is like measuring the average temperature in a hospital."

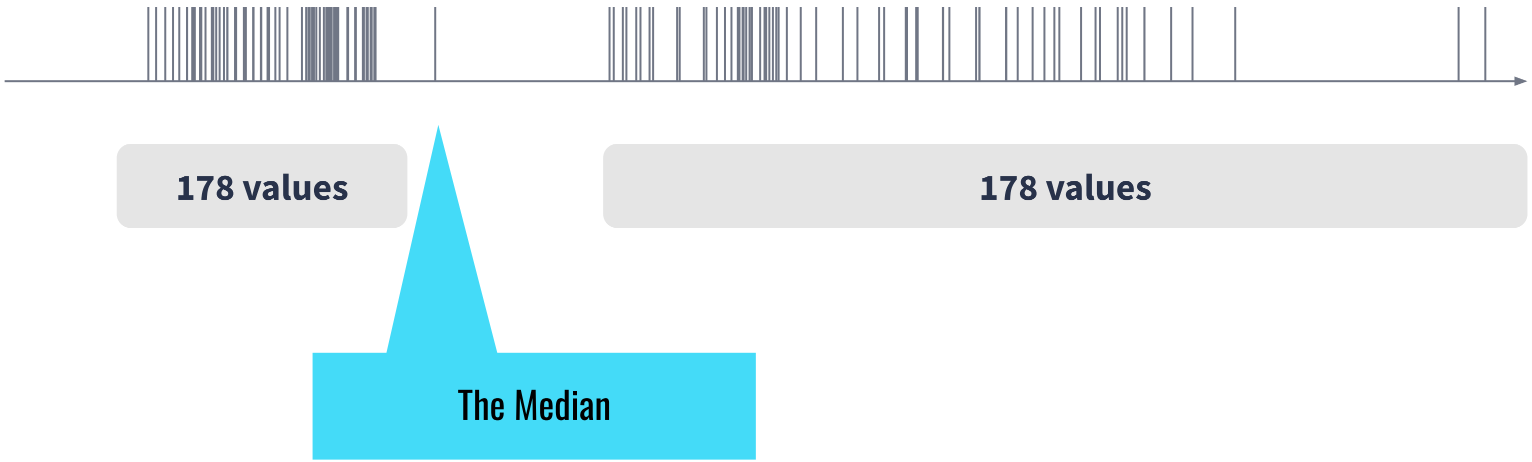
Source: Dogan Ugurlu @ Optimizely

Averages

- Most data we see on graphs is averaged: Be aware of Spike Erosion.
- Averages are easy to compute and mergeable
- Looking at average latency has little value

MEDIAN

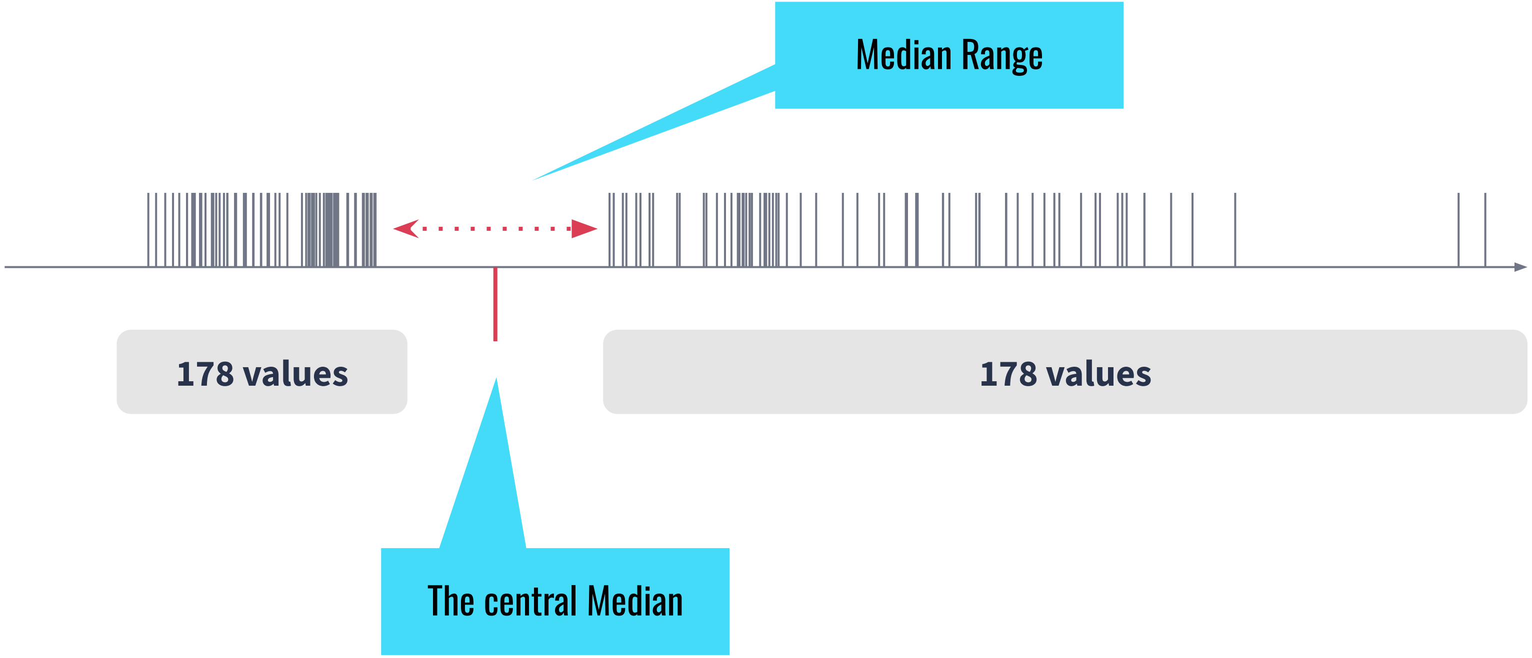
Median Definition



Median Definition

Where goes the Median?

Median Definition

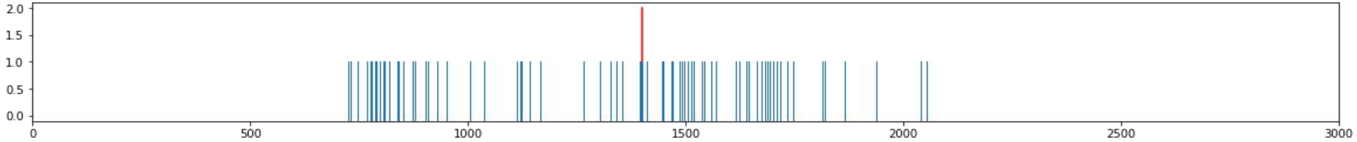


Summary Statistic - Medians

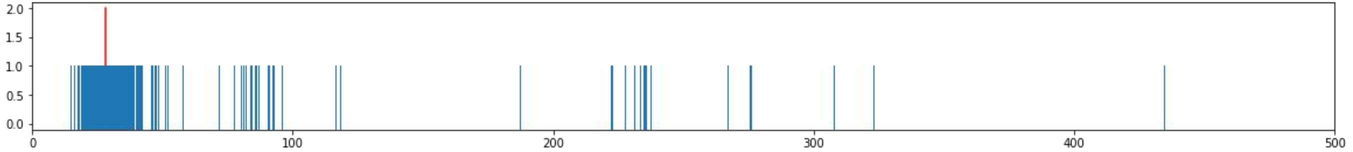
Median

Normal Noise

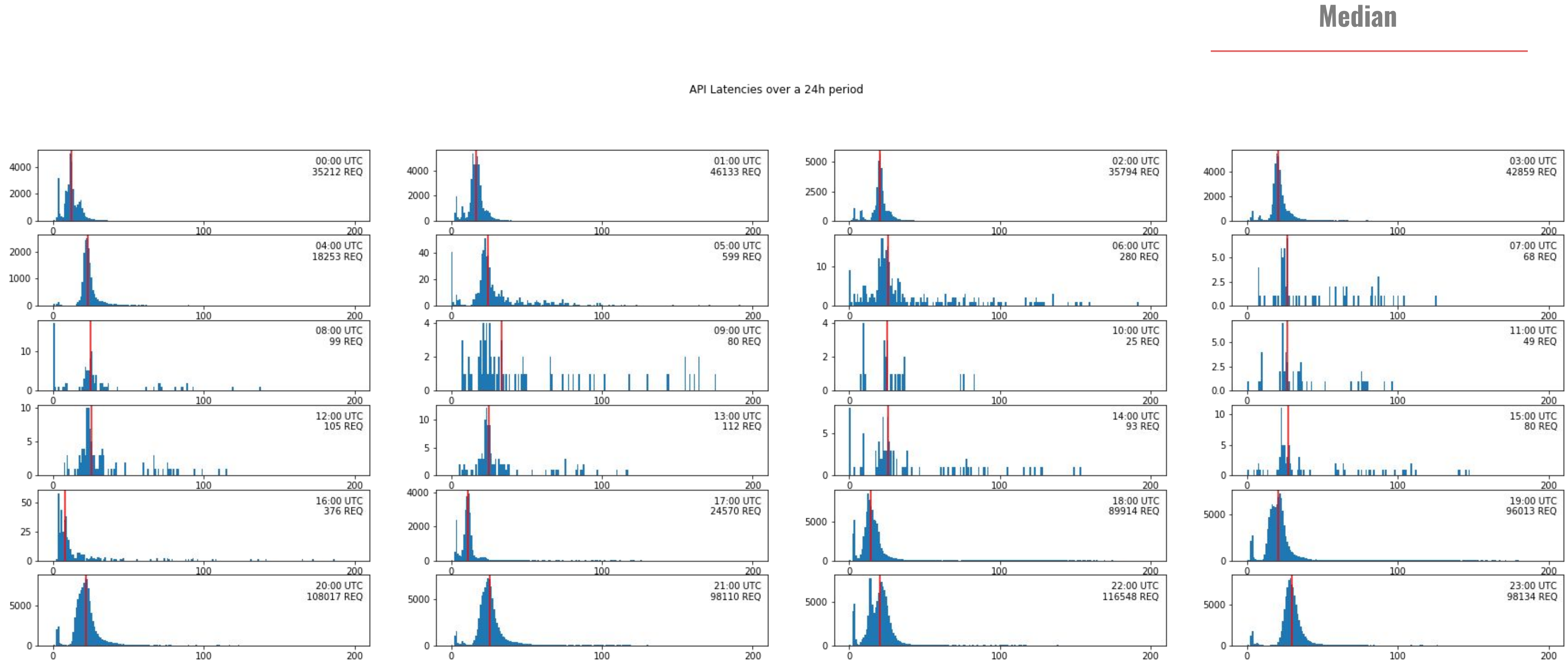
Request Rates



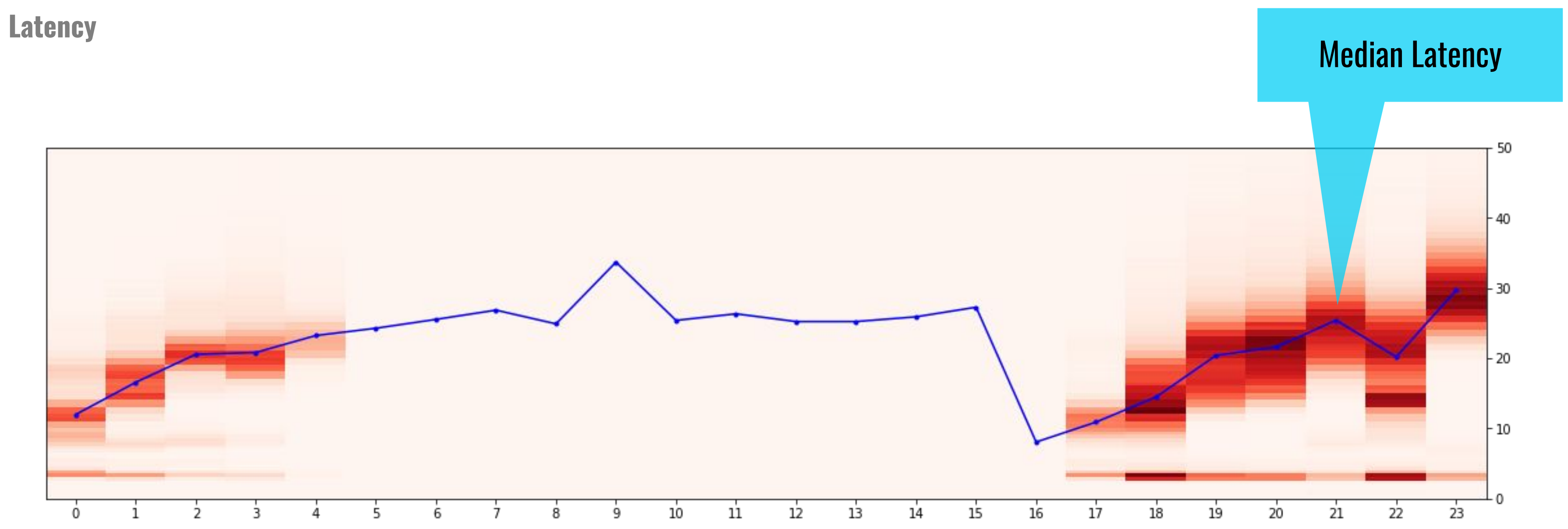
Request Latency



Example Dataset - Request Latencies over Time



Example Dataset - Request Latencies Mean Values

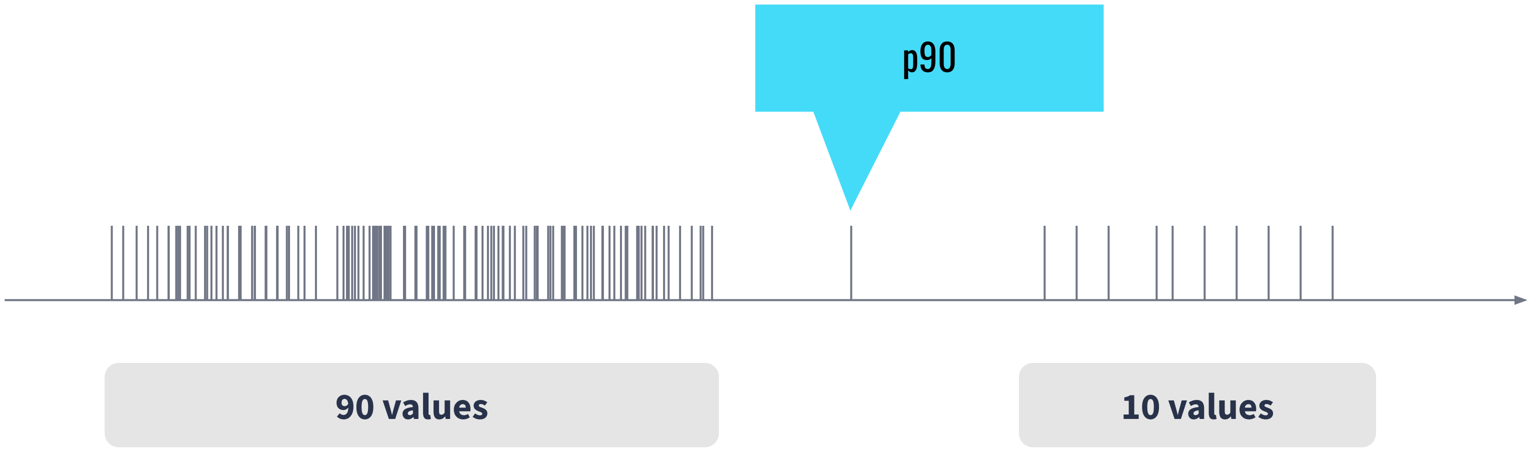


Median - Take Aways

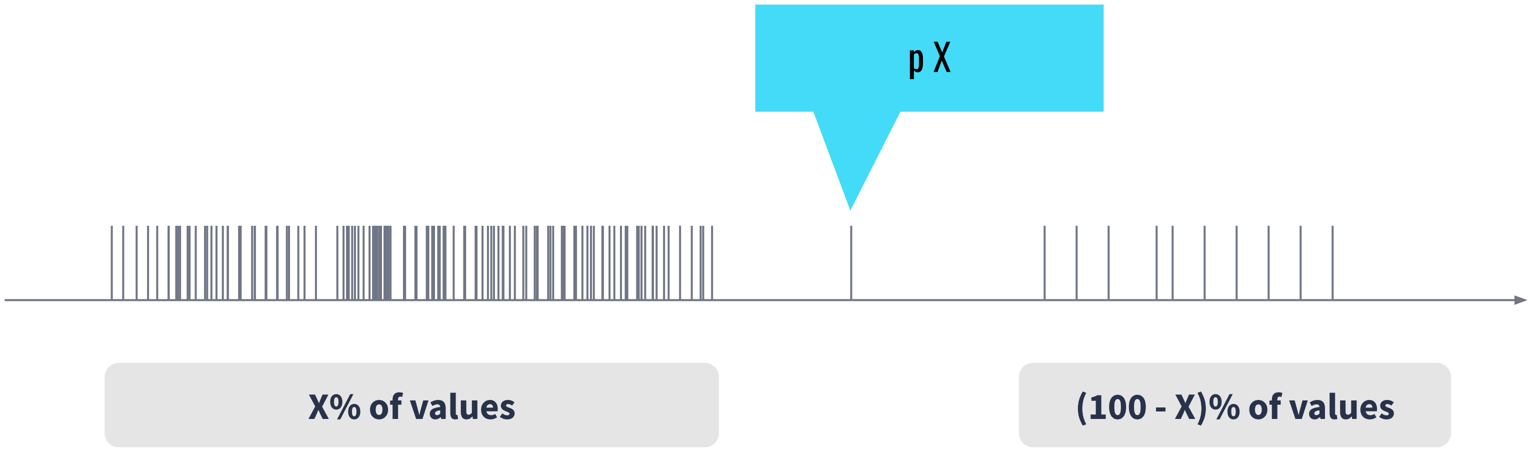
- Median values give "central representatives" of a data-set
- Medians are robust to outliers
- Fact: Medians are not easily mergeable

PERCENTILES

p90 Example



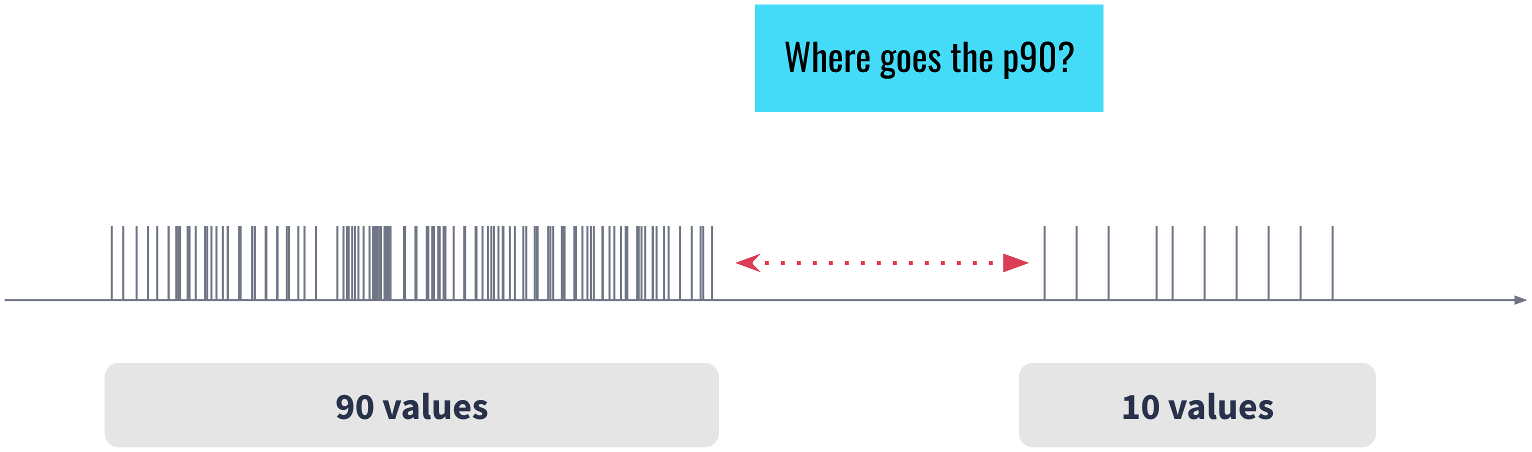
Percentile Definition



Percentile Special Cases

- ➔ p0 = Minimum
- ➔ p50 = Median
- ➔ p100 = Maximum

Percentile Definition



Lots of choices found in the wild

more on heinrichhartmann.com - Quantiles

The estimate types and interpolation schemes used include:

Type	h	Q_p
R-1, SAS-3, Maple-1	Np	$x_{\lceil h \rceil}$
R-2, SAS-5, Maple-2, Stata	$Np + 1/2$	$(x_{\lceil h-1/2 \rceil} + x_{\lfloor h+1/2 \rfloor}) / 2$
R-3, SAS-2	$Np - 1/2$	$x_{\lfloor h \rceil}$
R-4, SAS-1, SciPy-(0,1), Maple-3	Np	$x_{\lfloor h \rfloor} + (h - \lfloor h \rfloor)(x_{\lceil h \rceil} - x_{\lfloor h \rfloor})$
R-5, SciPy-(1/2,1/2), Maple-4	$Np + 1/2$	
R-6, Excel, Python, SAS-4, SciPy-(0,0), Maple-5, Stata-altdef	$(N+1)p$	
R-7, Excel, Python, SciPy-(1,1), Maple-6, NumPy, Julia	$(N-1)p+1$	
R-8, SciPy-(1/3,1/3), Maple-7	$(N+1/3)p + 1/3$	
R-9, SciPy-(3/8,3/8), Maple-8	$(N+1/4)p + 3/8$	

Wikipedia - Percentiles

Sample Quantiles in Statistical Packages

Rob J. HYNDMAN and Yanan FAN

There are a large number of different definitions used for sample quantiles in statistical computer packages. Often within the same package one definition will be used to compute a quantile explicitly, while other definitions may be used when producing a boxplot, a probability plot, or a QQ plot. We compare the most commonly implemented sample quantile definitions by writing them in a common notation and investigating their motivation and some of their properties. We argue that there is a need to adopt a standard definition for sample quantiles so that the same answers are produced by different packages and within each package. We conclude by recommending that the median-unbiased estimator be used because it has most of the desirable properties of a quantile estimator and can be defined independently of the underlying distribution.

KEY WORDS: Percentiles; Quartiles; Sample quantiles; Statistical computer packages.

1. INTRODUCTION

The quantile of a distribution is defined as

$$Q(p) = F^{-1}(p) = \inf\{x: F(x) \geq p\}, \qquad 0 < p < 1,$$

where $F(x)$ is the distribution function. Sample quantiles provide nonparametric estimators of their population counterparts based on a set of independent observations $\{X_1, \ldots, X_n\}$ from the distribution F. Let $\{X_{(1)}, \ldots, X_{(n)}\}$ denote the order statistics of $\{X_1, \ldots, X_n\}$, and let $\hat{Q}_i(p)$ denote the ith sample quantile definition.

One difficulty in comparing quantile definitions is that there is a number of equivalent ways of defining them. However, the sample quantiles that are used in statistical packages are all based on one or two order statistics, and can be written as

$$\hat{Q}_i(p) = (1-\gamma)X_{(j)} + \gamma X_{(j+1)}$$

$$\text{where} \quad \frac{j-m}{n} \leq p < \frac{j-m+1}{n} \qquad (1)$$

for some $m \in \mathbb{R}$ and $0 \leq \gamma \leq 1$. The value of γ is a function of $j = \lfloor pn + m \rfloor$ and $g = pn + m - j$. Here, $\lfloor u \rfloor$ denotes the largest integer not greater than u; later we shall use $\lceil u \rceil$ to denote the smallest integer not less than u.

We consider estimators of the form (1), including some that are not found in statistical packages. There have been several other nonparametric quantile estimators proposed that are not of the form (1) (e.g., Harrell and Davis 1982; Sheather and Marron 1990), but these are not implemented in widely available packages and so are not considered here. We also exclude sample quantiles that are not defined for all p including hinges and other letter values (Hoaglin 1983) and related methods (Freund and Perles 1987).

A closely related problem is the selection of plotting position in a quantile plot in which $X_{(k)}$ is plotted against p_k or in a quantile–quantile plot in which $X_{(k)}$ is plotted against $G^{-1}(p_k)$ where G is a distribution function. Various rules for p_k have been suggested (see Cunnane 1978; Harter 1984; Kimball 1960; Mage 1982). Each plotting rule corresponds to a sample quantile definition by defining $\hat{Q}_i(p_k) = X_{(k)}$ and using linear interpolation for $p \neq p_k$. However, the criteria by which a plotting position is chosen (e.g., the five postulates of Gumbel 1958, pp. 32–34 or the three purposes of Kimball 1960) may be quite different from the criteria for choosing a good sample quantile definition.

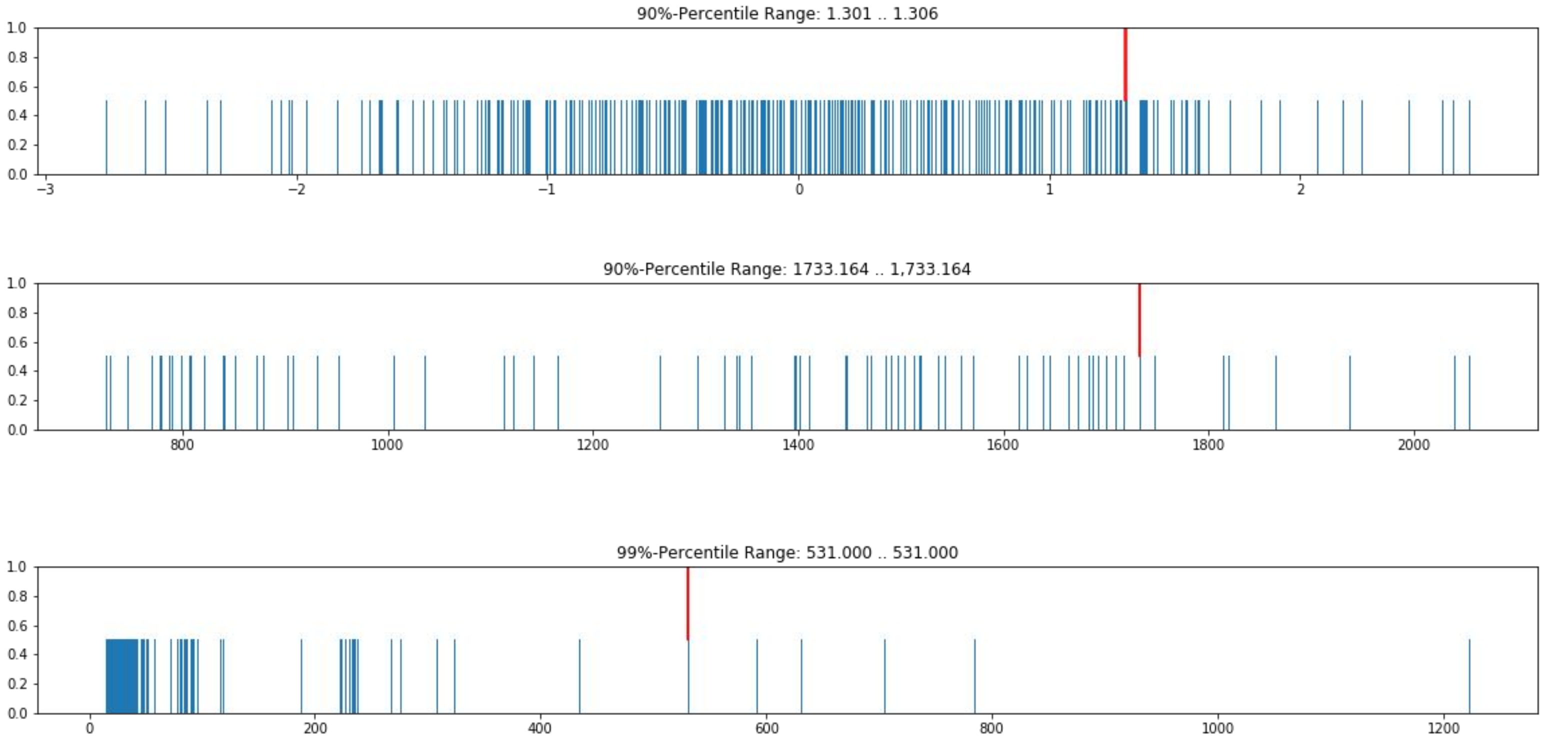
We compare sample quantile definitions of the form (1) by describing their motivation and whether or not they pos-

Table 1. Six Desirable Properties for a Sample Quantile

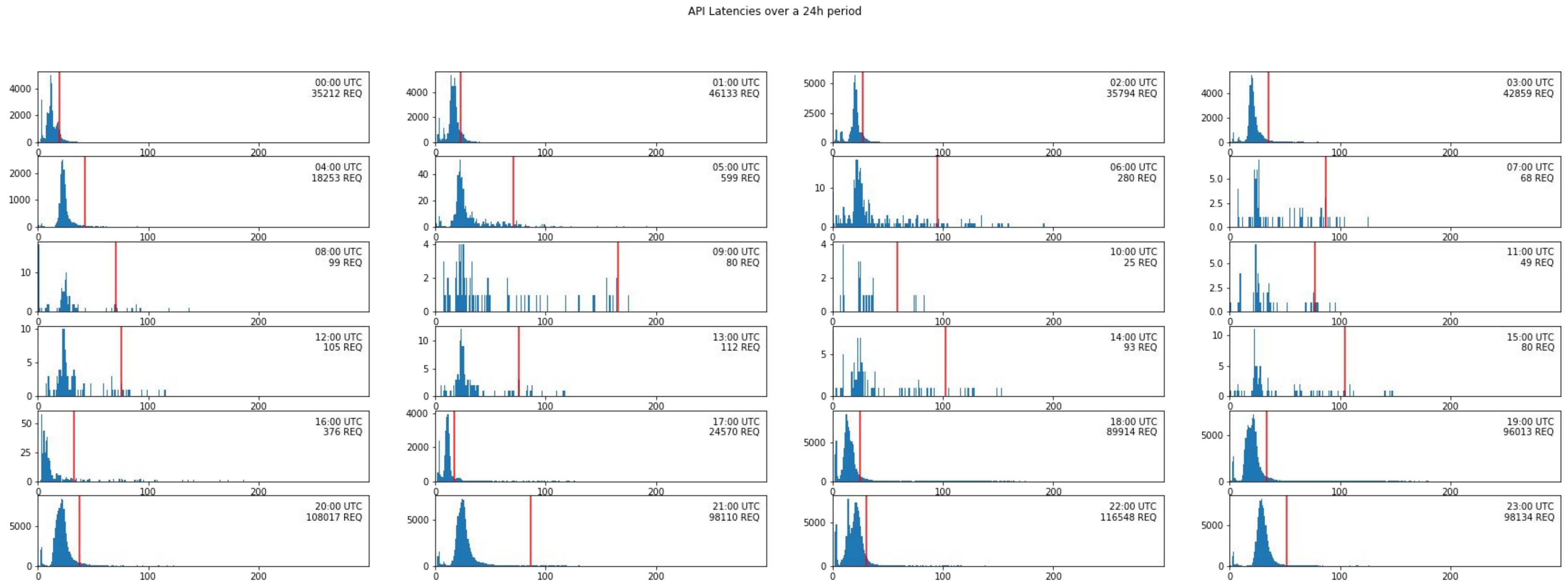
P1: $\hat{Q}_i(p)$ is continuous.
P2: $\text{Freq}(X_k \leq \hat{Q}_i(p)) \geq pn$.
P3: $\text{Freq}(X_k \leq \hat{Q}_i(p)) = \text{Freq}(X_k \geq \hat{Q}_i(1-p))$.
P4: Where $\hat{Q}_i^{-1}(x)$ is uniquely defined,

Hyndman, Fan (1996)

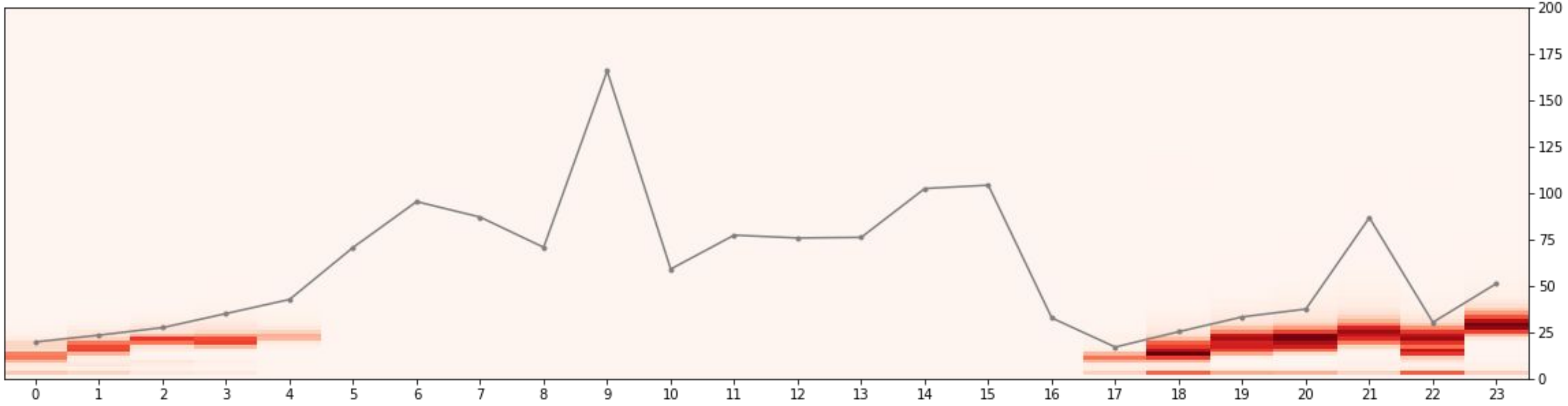
Summary Statistic - Percentile



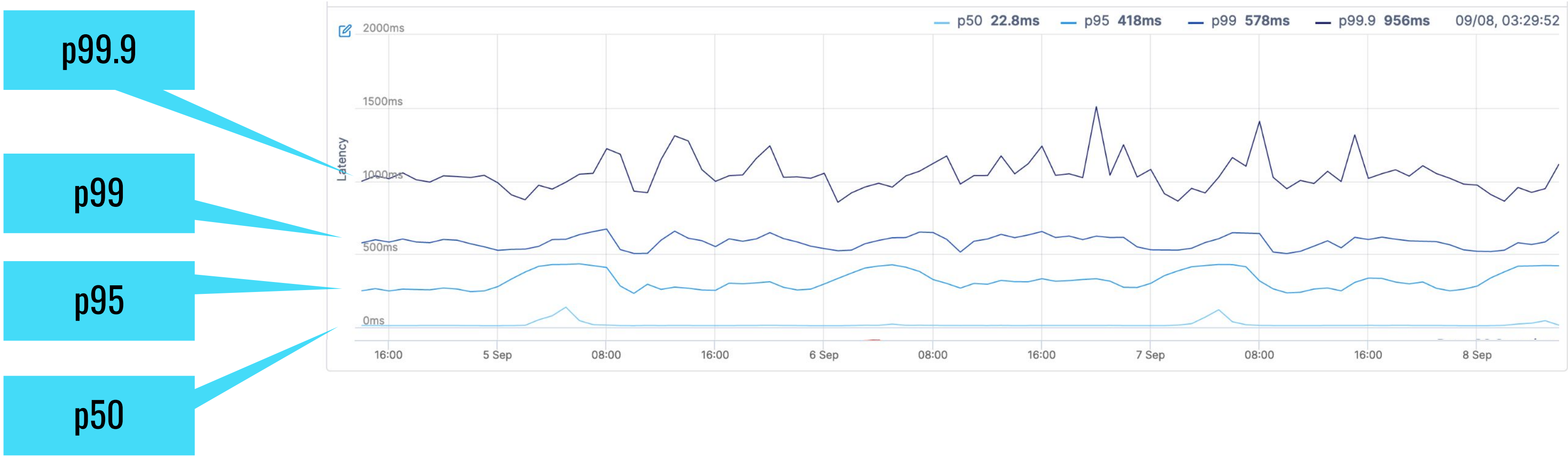
Latency Percentiles



Latency Percentiles - over time

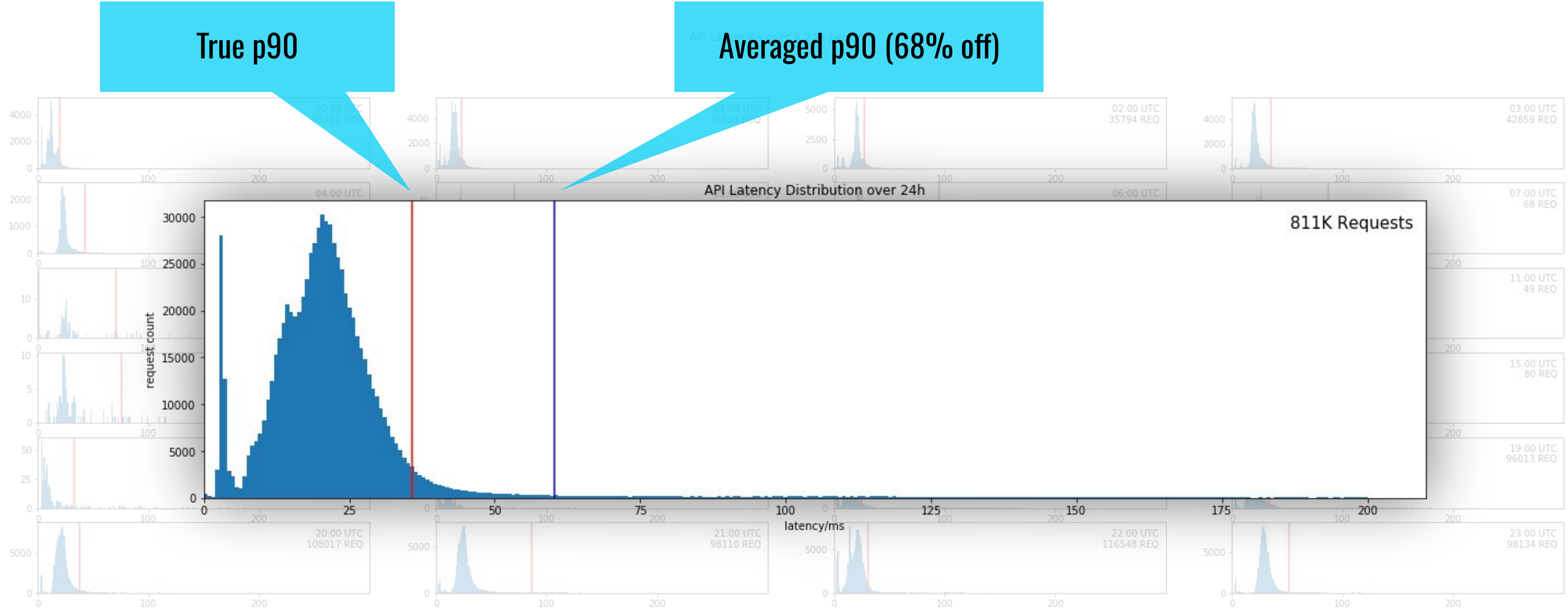


Example - Production View - Request Latency over Time



Lightstep from ServiceNow

Aggregated Percentiles

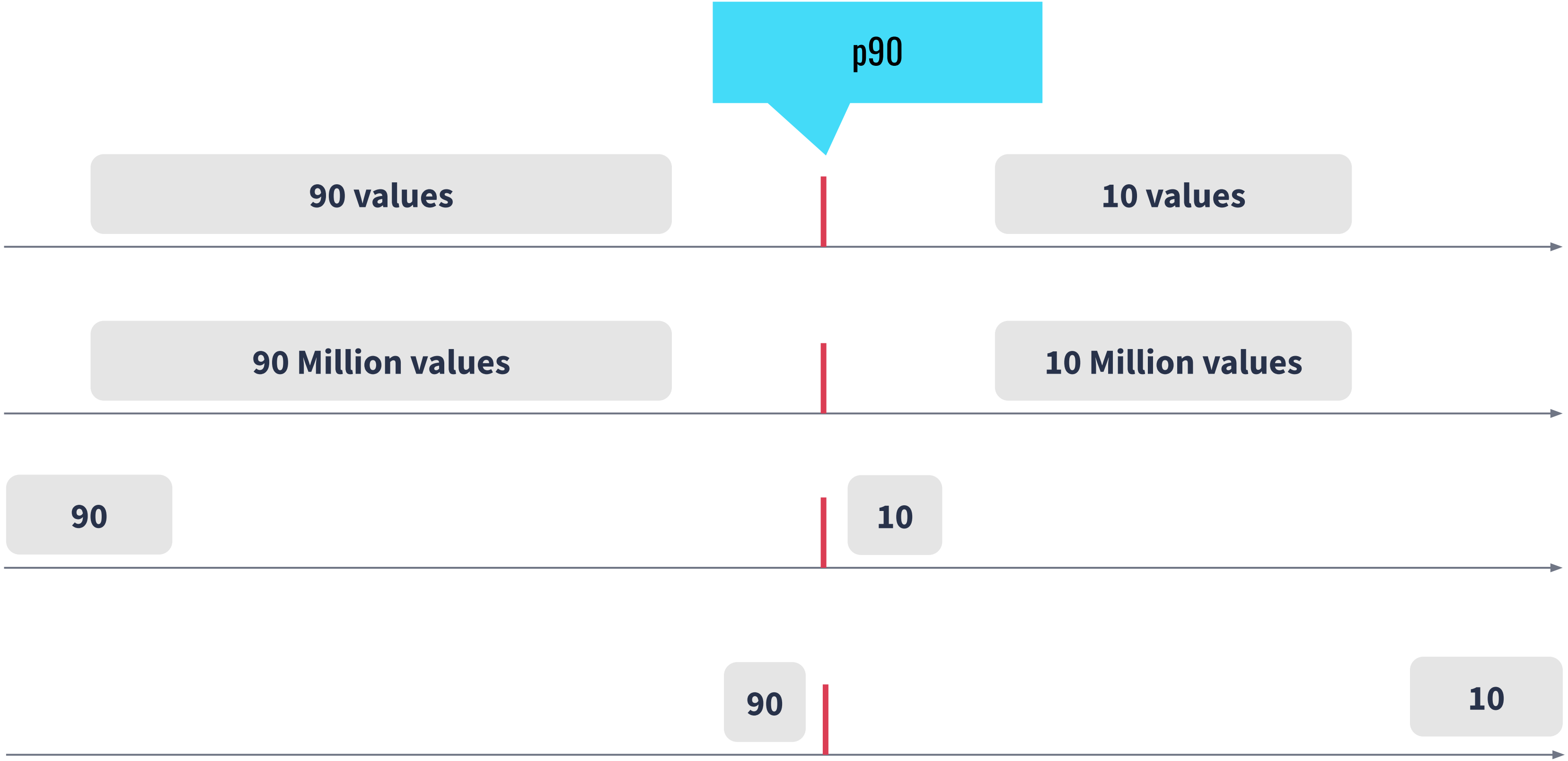


“

Don't average percentiles!

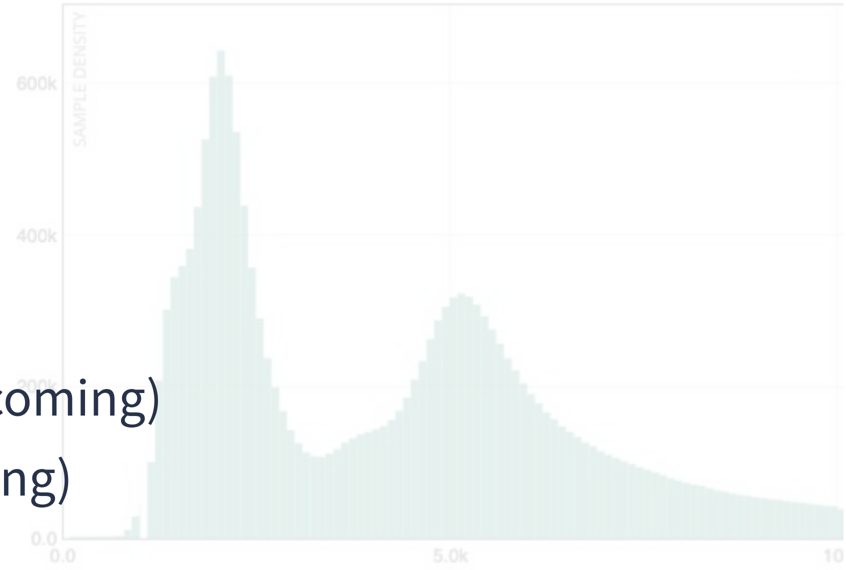
(... and expect the results to be something meaningful)

Distributions with the same p90



How to do better? Use Histogram data-structures!

- HDR Histogram
- T-Digest
- OpenHistogram (Circllhist)
- DD/UUD-Sketch
- Prometheus Sparse Histograms (upcoming)
- OpenTelemetry Histograms (upcoming)



Percentiles - Take Aways

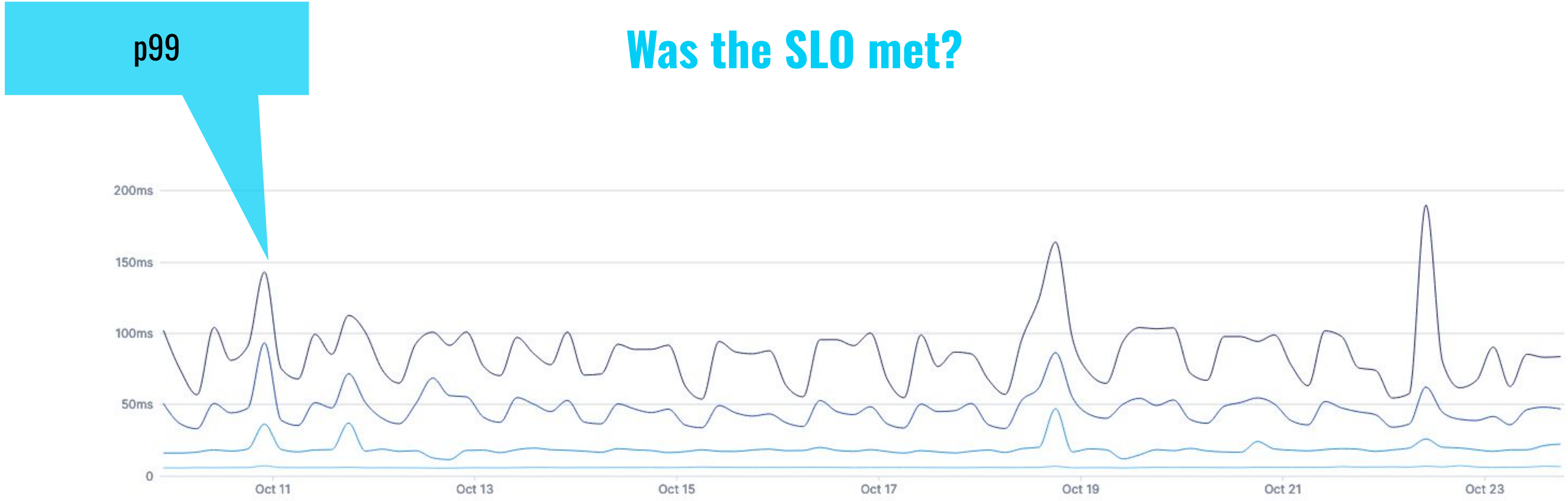
- Percentiles generalize min/max/median
- Percentiles (p90, p99, p99.9) are used to describe latency
- Percentiles can't be aggregated (need histograms for this!)

#2 Latency SLOs

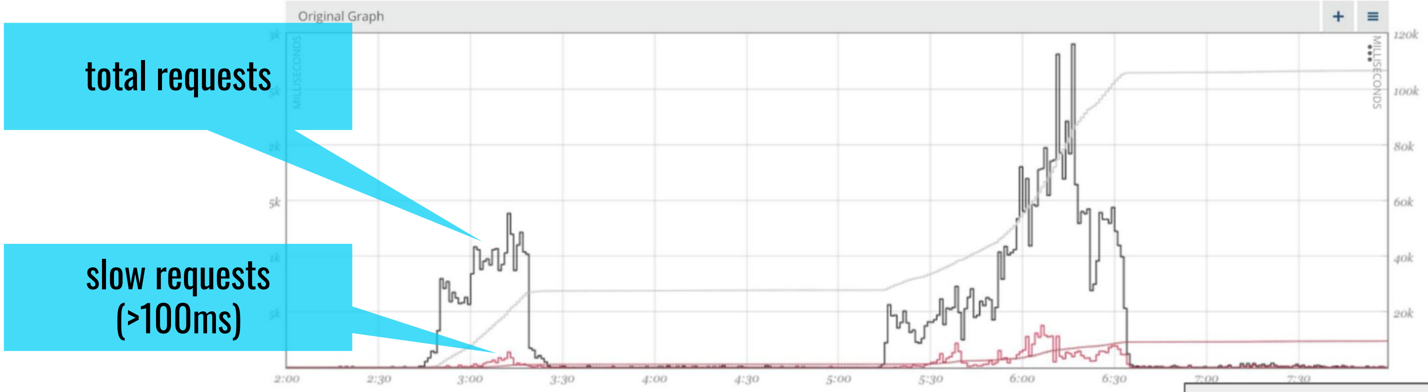
Latency SLO

99% of the requests in the past 28 days were served within 100ms.

Was the SLO met?



Option #1 - Counter Metrics



	Jan 28 2019, 08:00 (1M)
Total Request Count (ms)	0k
Total Requests Cumulative (ms)	106.599k
Bad Request Count (Slower than 50ms) (ms)	0k
Bad Requests Cumulative (ms)	9.534k

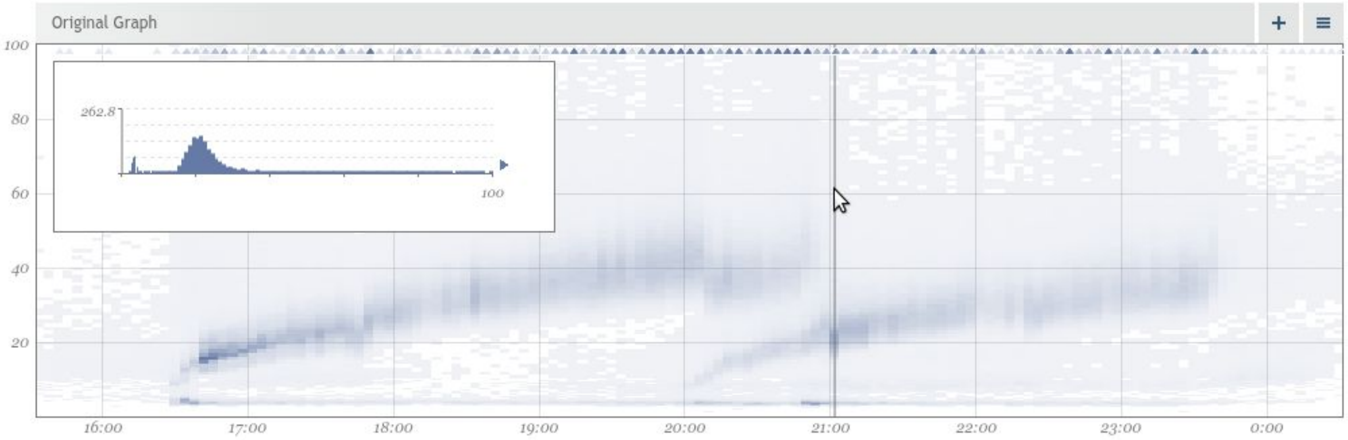
9.5 K / 106 K = 8.9%

Slow Requests

CIRCONUS

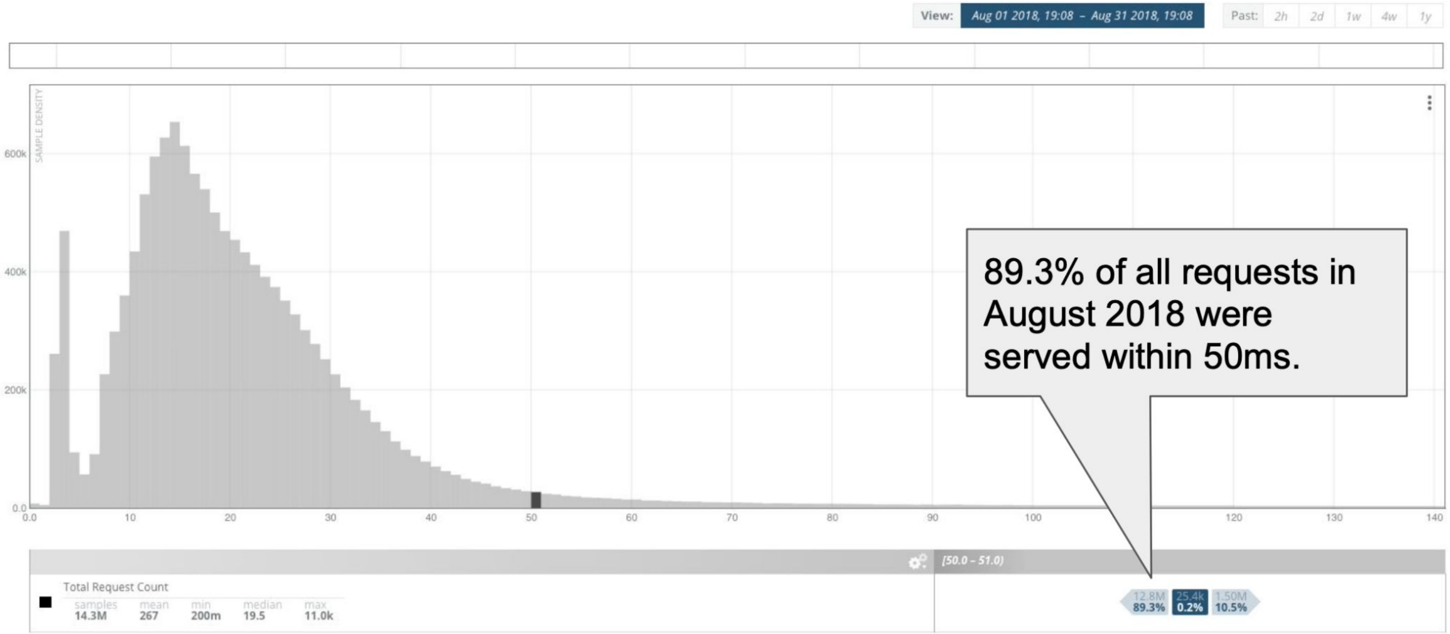
CIRCONUS

Option #2 - Histograms



CIRCONUS

Option #2 - Histograms

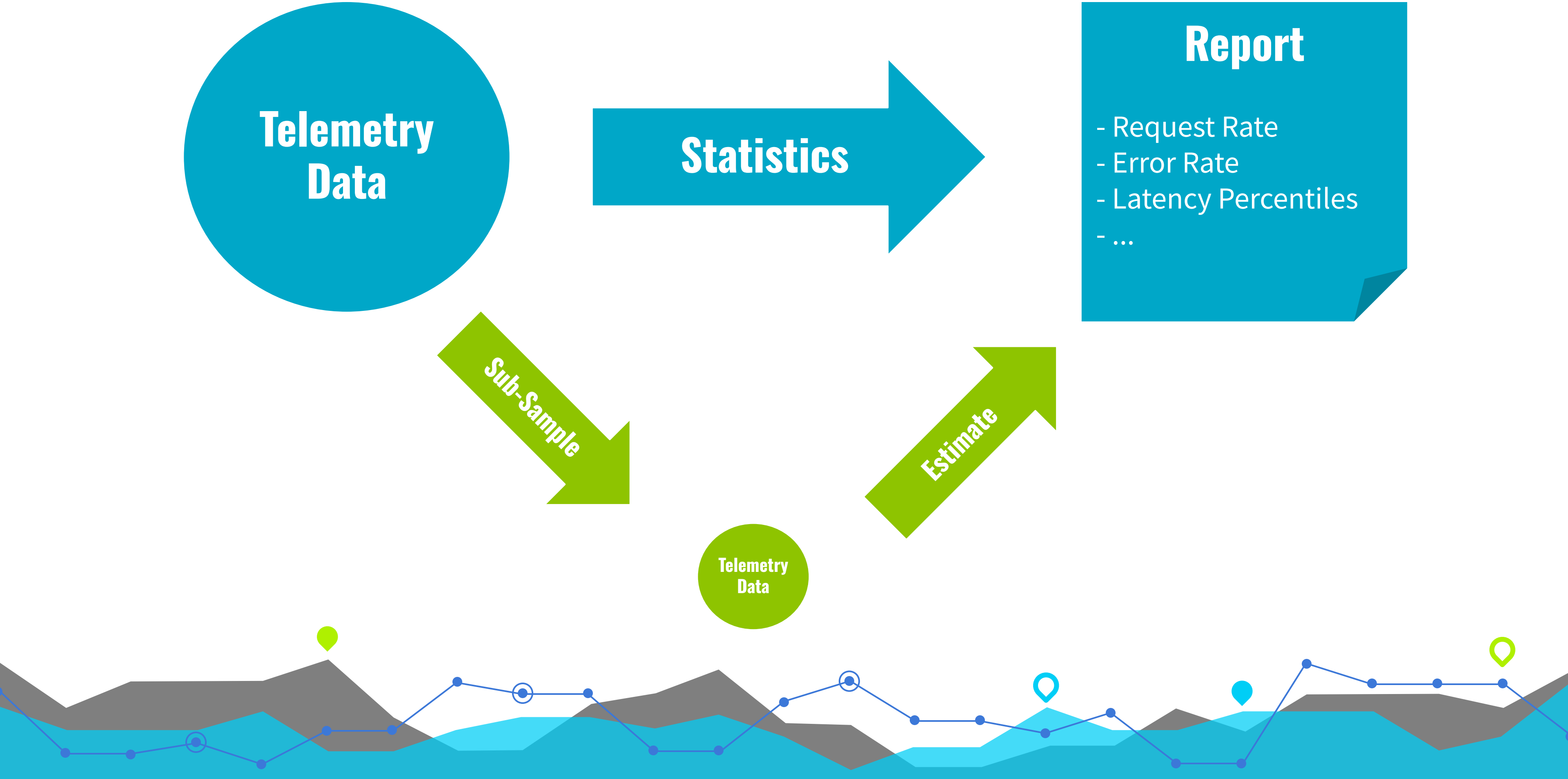


#3 Sampling

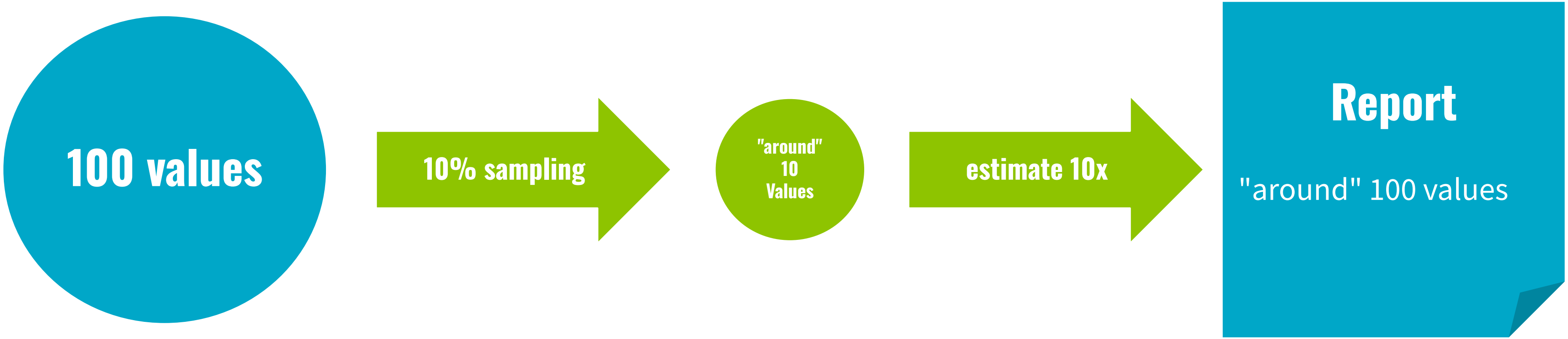
Accuracy
Graphs, Dashboards, KPIs, SLOs

Sampling

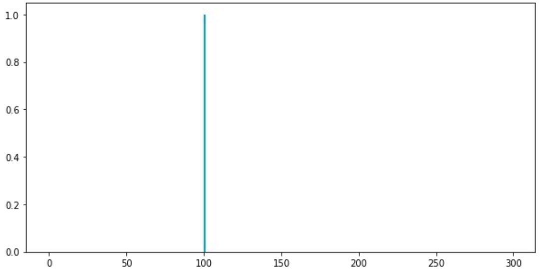
Costs
Logs/Metrics/Traces

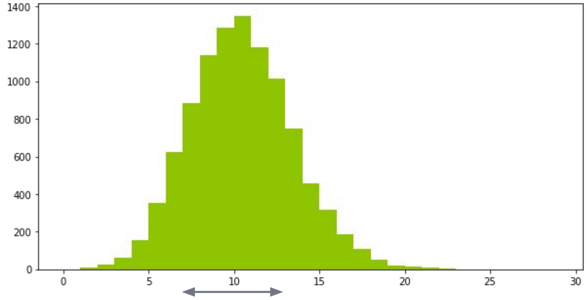

Telemetry Data
Statistics
Report
- Request Rate
- Error Rate
- Latency Percentiles
- ...
Sub-Sample
Estimate
Telemetry Data

Sampling

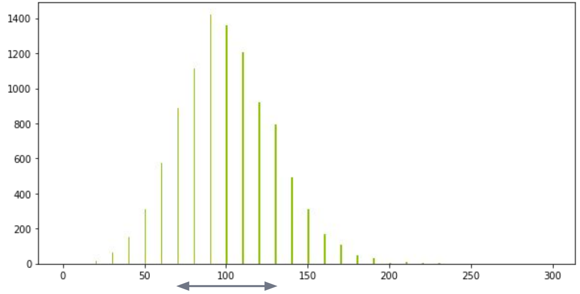

100 values
10% sampling
"around"
10
Values
estimate 10x
Report
"around" 100 values

Sampling - Simulation (10K iterations)



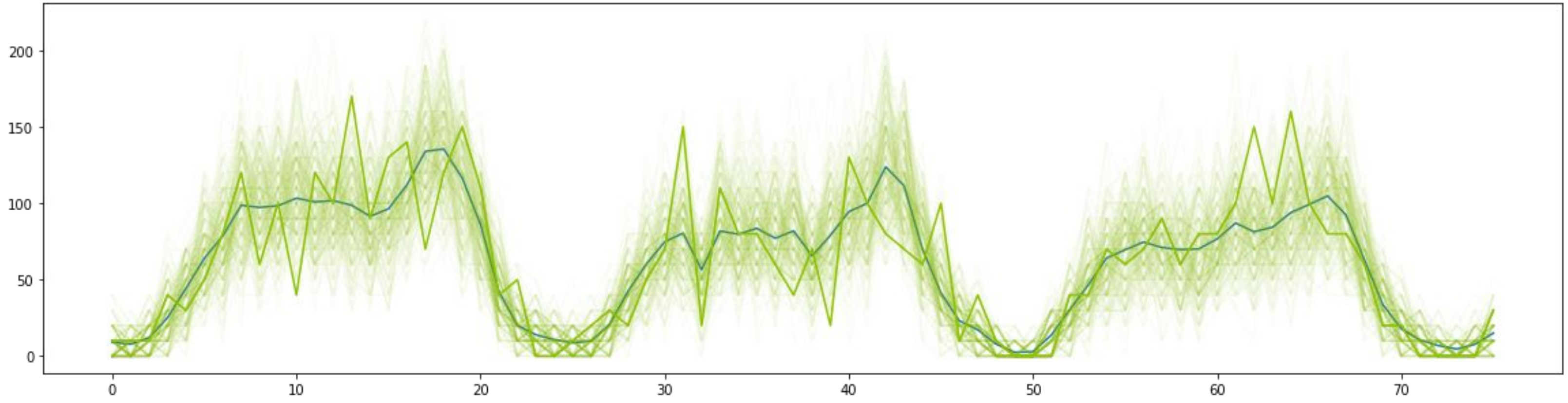


Measured stddev: 2.98

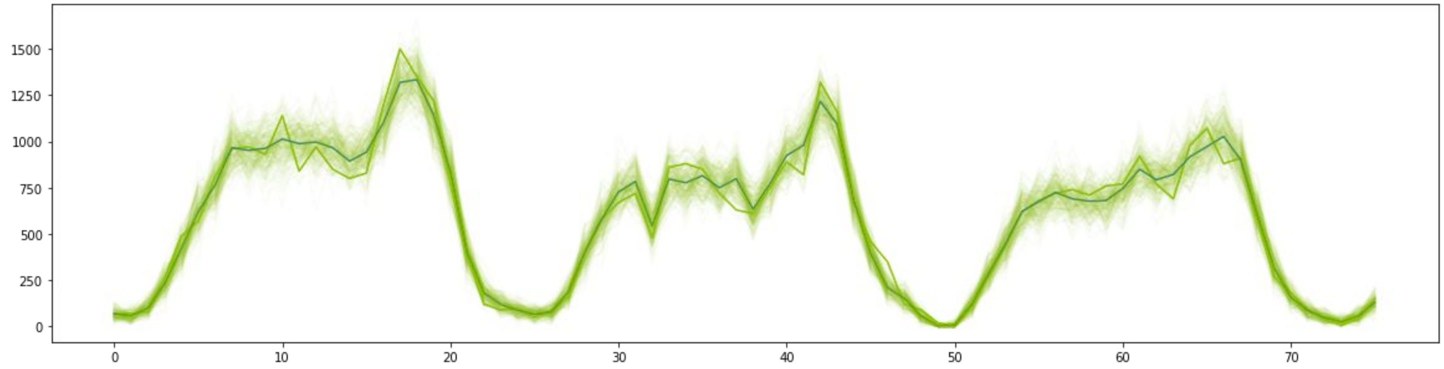


Measured stddev: 29.8

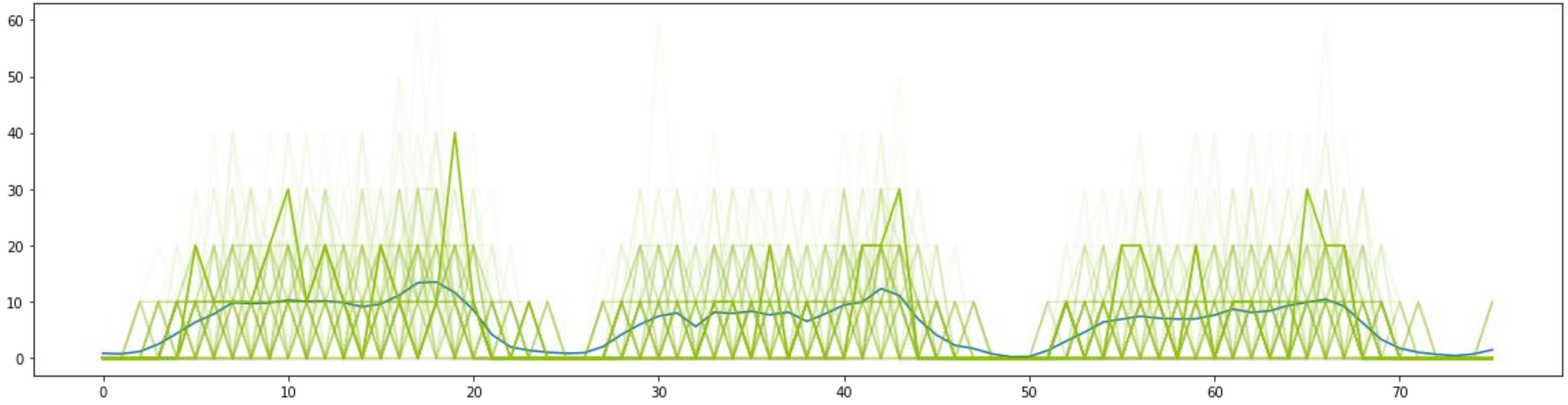
Sampling - Request Rates



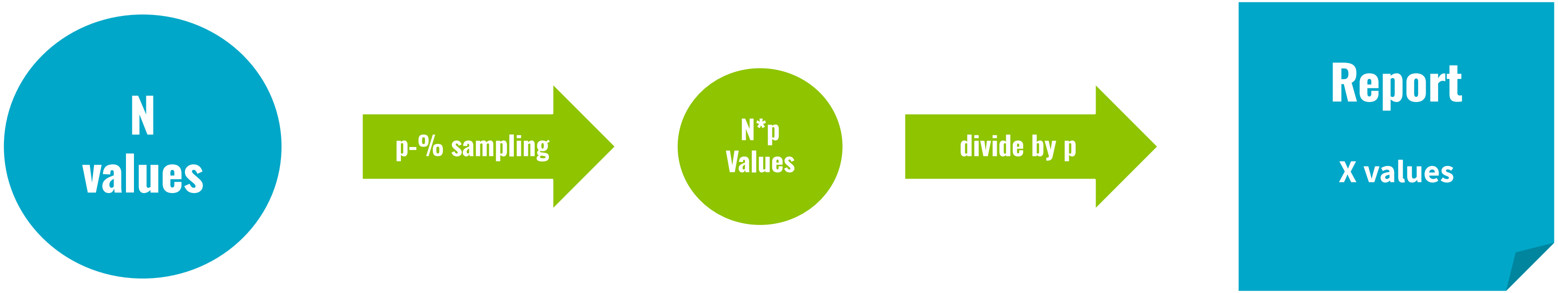
Sampling - Request Rates



Sampling - Request Rates



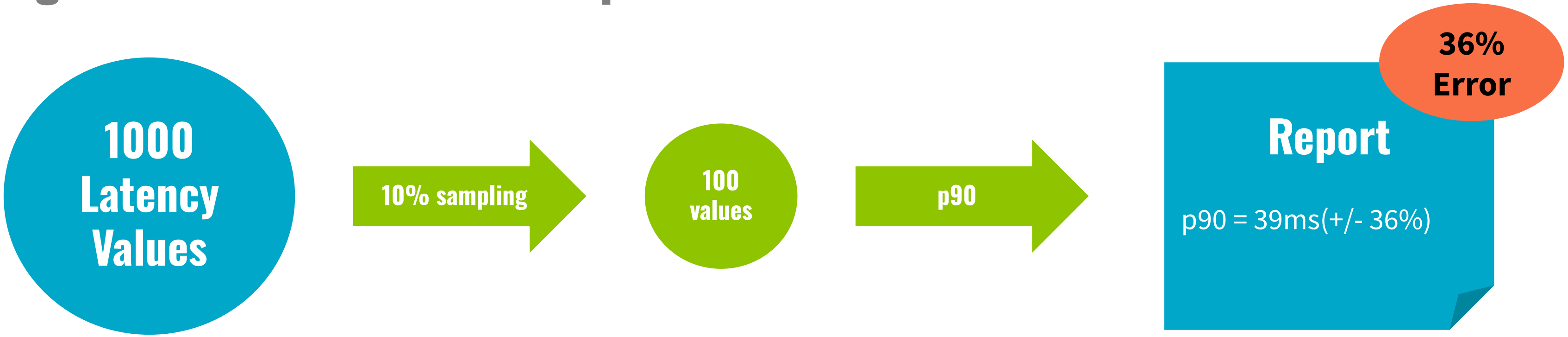
Sampling - Theory

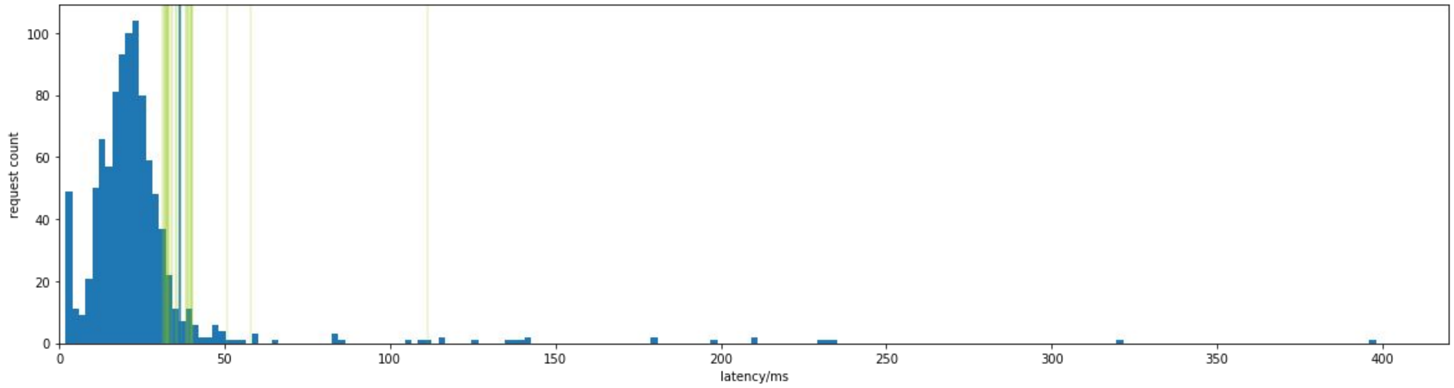


Proposition

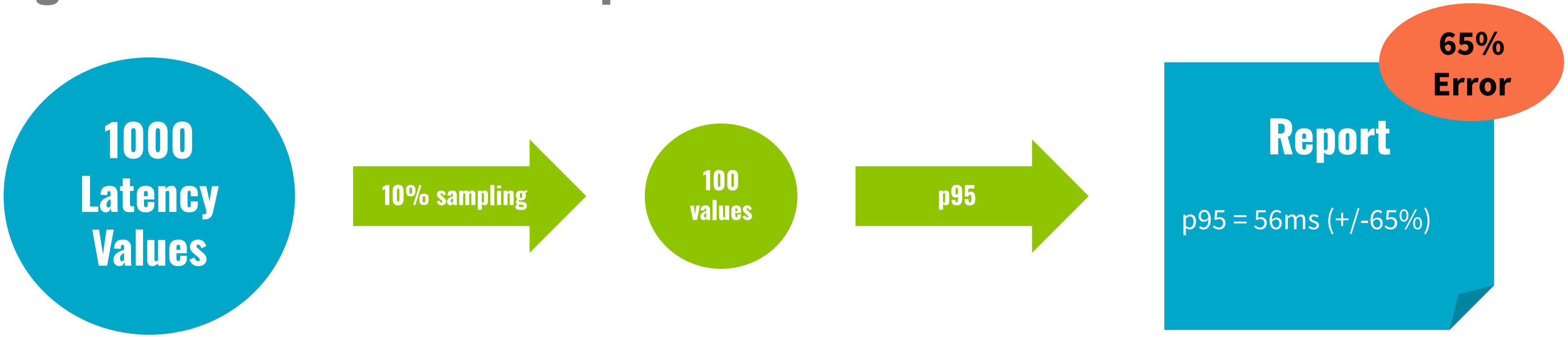
- X is a random variable following a scaled binomial distribution
- The expected value of X is N
- The standard deviation of X is SQRT((1 - p) / (N * p))

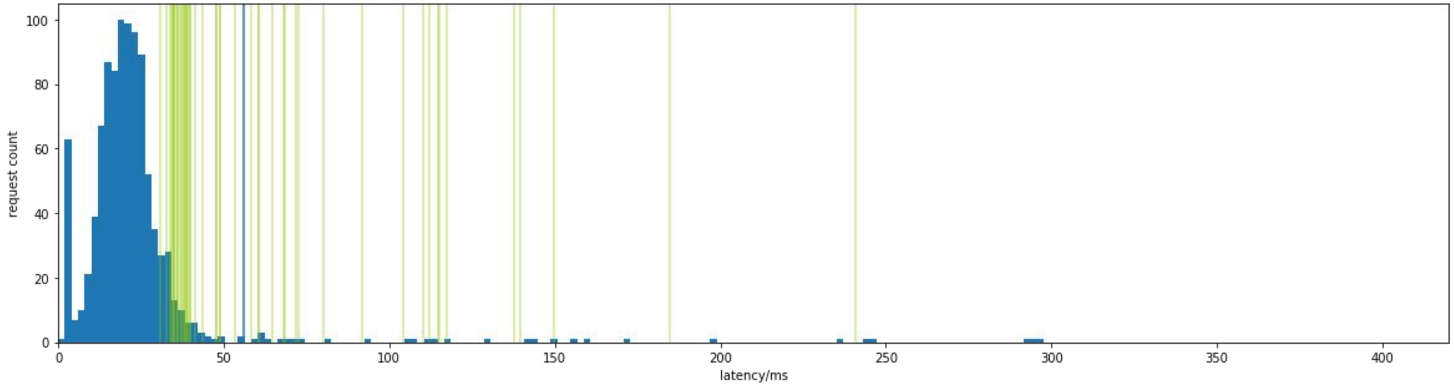
Sampling - Simulation - Percentiles p90



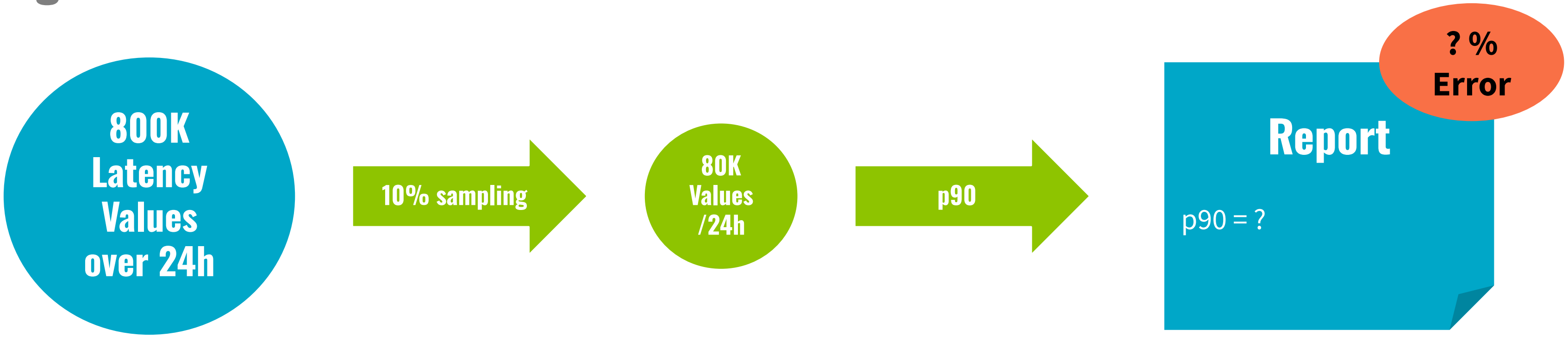


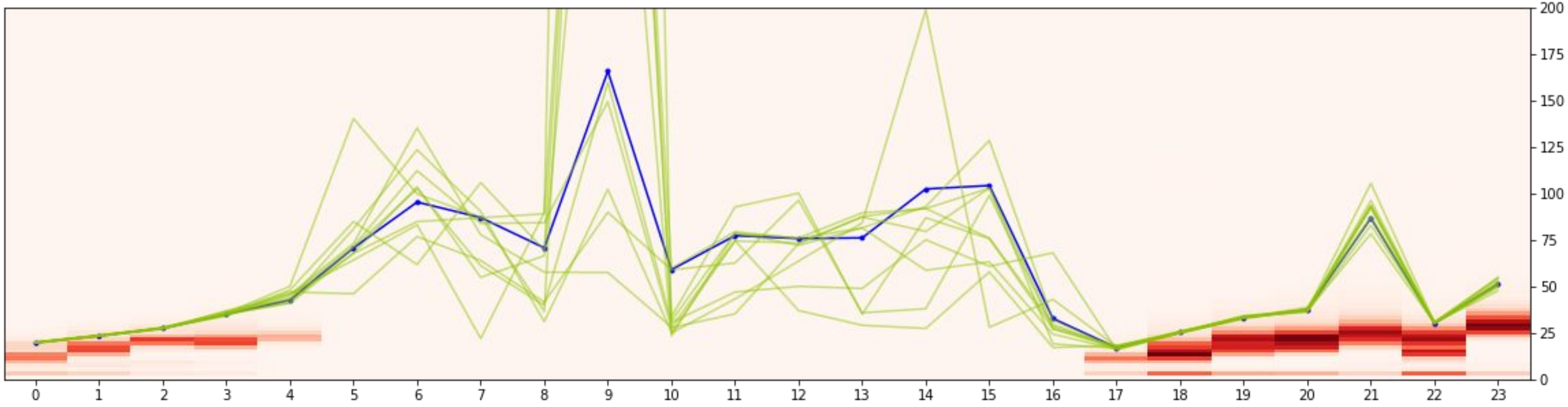
Sampling - Simulation - Percentiles p95





Sampling - Simulation - Percentiles





Sampling - Take Aways

- Effective way to trade costs vs. accuracy
- Accuracy loss depends on sample size and other factors
- Simulation gives effective tool to study sampling impact

The Sampling Error Calculator

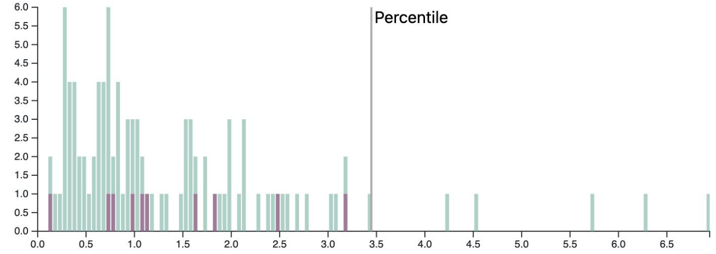
heinrichhartmann.com/sampling

# Sampling			
Sampling Rate	10 %		

# Request Rates			
Request Rate	100 rpm		
Time window	1 min (60 sec)		
Population	Total 100 requests contained in 60 sec time-window.		
## Samping Effects on Request Rate Estimates			1000 iterations
Sample	We expect to retrain 10 requests, after sampling with 10% probability.		
	Value	Standard Error	Realtive Error
Estimate Req. Count	100.0 req	± 30.00 req	30.00%
Simulate Req. Count	98.3 req	± 30.31 req	30.82%
Estimate Req. Rate	1.7 rps	± 0.50 rps	30.00%
Simulate Req. Rate	1.6 rps	± 0.51 rps	30.82%

# Error Rates			
Error Rate	5 %		
Population	From the 100 requests 5 are marked as error.		
## Samping Effects on Error Rate Estimates			1000 iterations
Sample	We expect to retrain 0.5 errors in the sample of size 10.		
	The probability that no error will be retained is 59.049000000%		
Estimate Err. Rate	5.0 %	± 7.02 ppt	140.36%
Simulate Err. Rate	4.9 %	± 7.33 ppt	150.90%

# Latency			
Latency Distribution	LogNormal		
Percentile p 95.0	3.4 ms		
Population	Total 100 requests following LogNormal distribution.		
	The true p95.0 is at 3.4ms.		
## Samping Effects on Latency			1000 iterations
Sample	We expect to retain 10.0 requests.		
Estimate Percentile p95.0	2.78 ms	± 1.68 ms	60.45%



Thank You!

Further Reading

- twitter: @HenrichHartmann

- blog: heinrichhartmann.com / sampling

- source: github.com/HeinrichHartmann/Statistics-for-Engineers