Advanced Systems Lab
Tutorial III
Statistics and Analysis

G. Alonso
Systems Group
http://www.systems.ethz.ch

Reading assignment

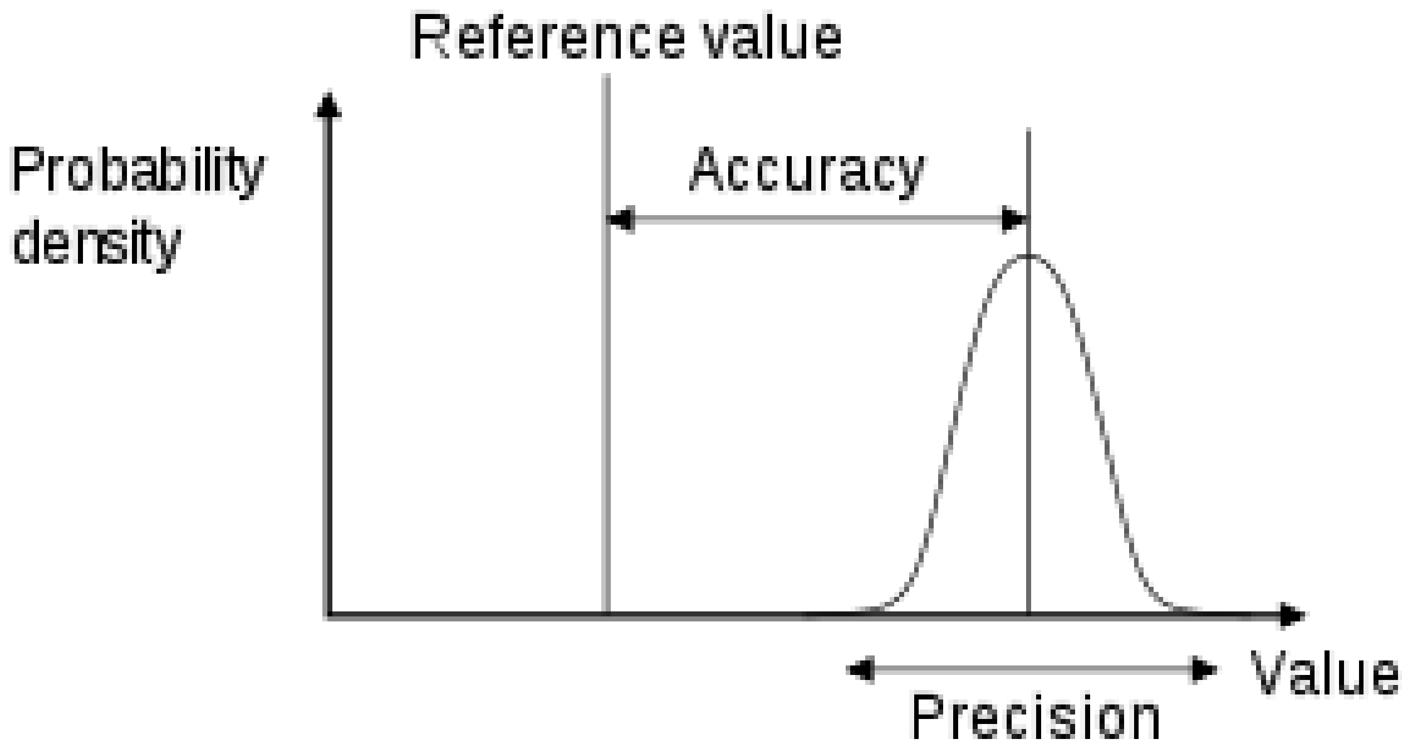
- Read chapters 10, 11, 12, and 13
- Read chapters 17 to 22

Basic statistics

- Not a course on statistics
  - You have done that already
  - We assume familiarity with the basics
- Focus on experimental aspects
  - What and when to measure
  - Side effects and different performance patterns
  - Data distributions
  - Sampling
  - Mean, Average, Outliers, deviation, plotting
  - Confidence intervals

Accuracy vs. Precision

Accuracy = how close to the real value (often unknown)
Precision = similarity of the results of repeated experiments



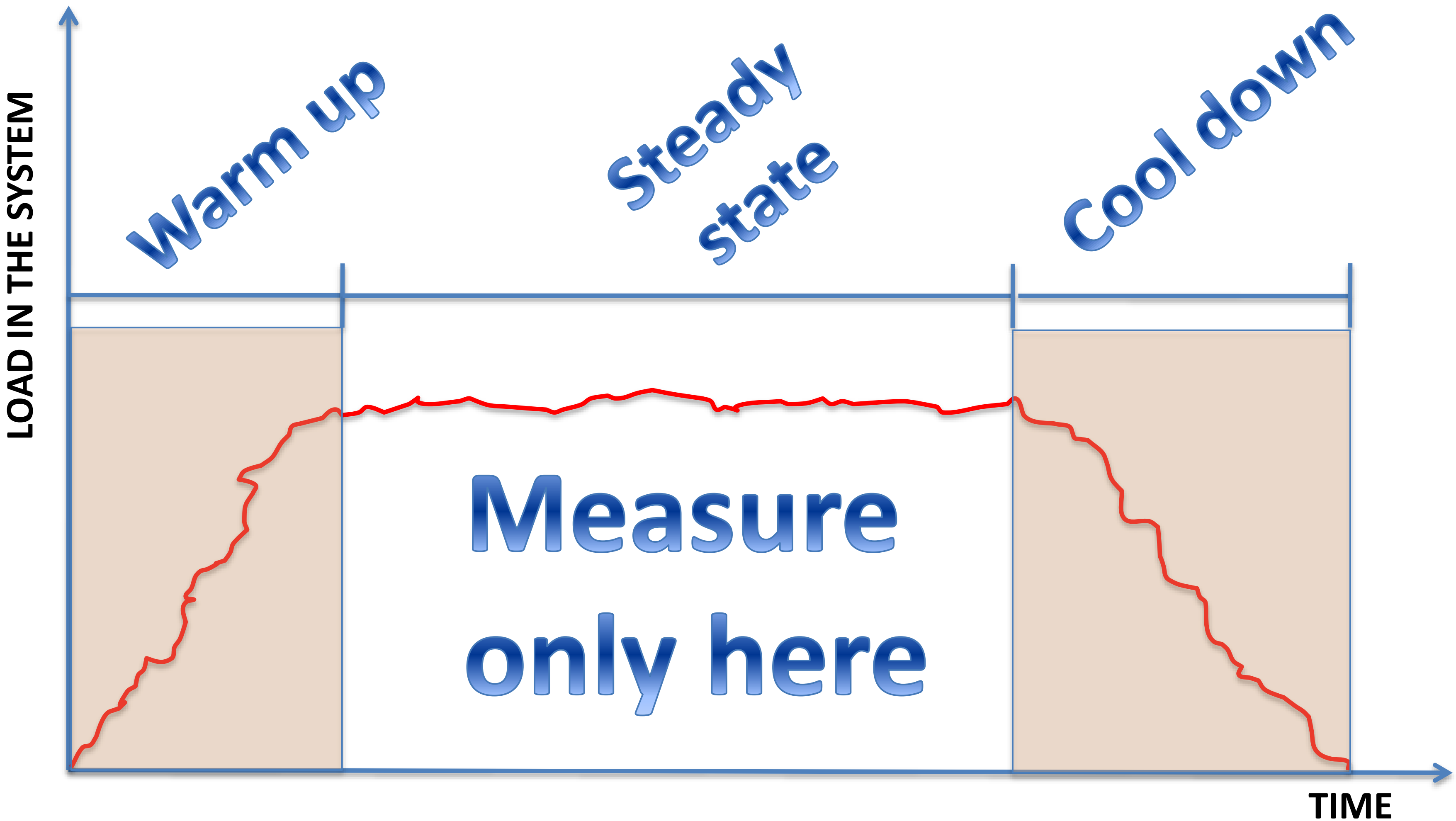
http://en.wikipedia.org/wiki/Accuracy_and_precision

When to measure

What and when to measure

- Decide on the parameters to measure:
  - Throughput, response time, latency, etc.
- Design your experiment
  - Configuration, data, load generators, instrumentation, hypothesis
- Run the experiment and start measuring:
  - When to measure (life cycle of an experiment)
  - What to measure (sampling)

Life cycle of an experiment



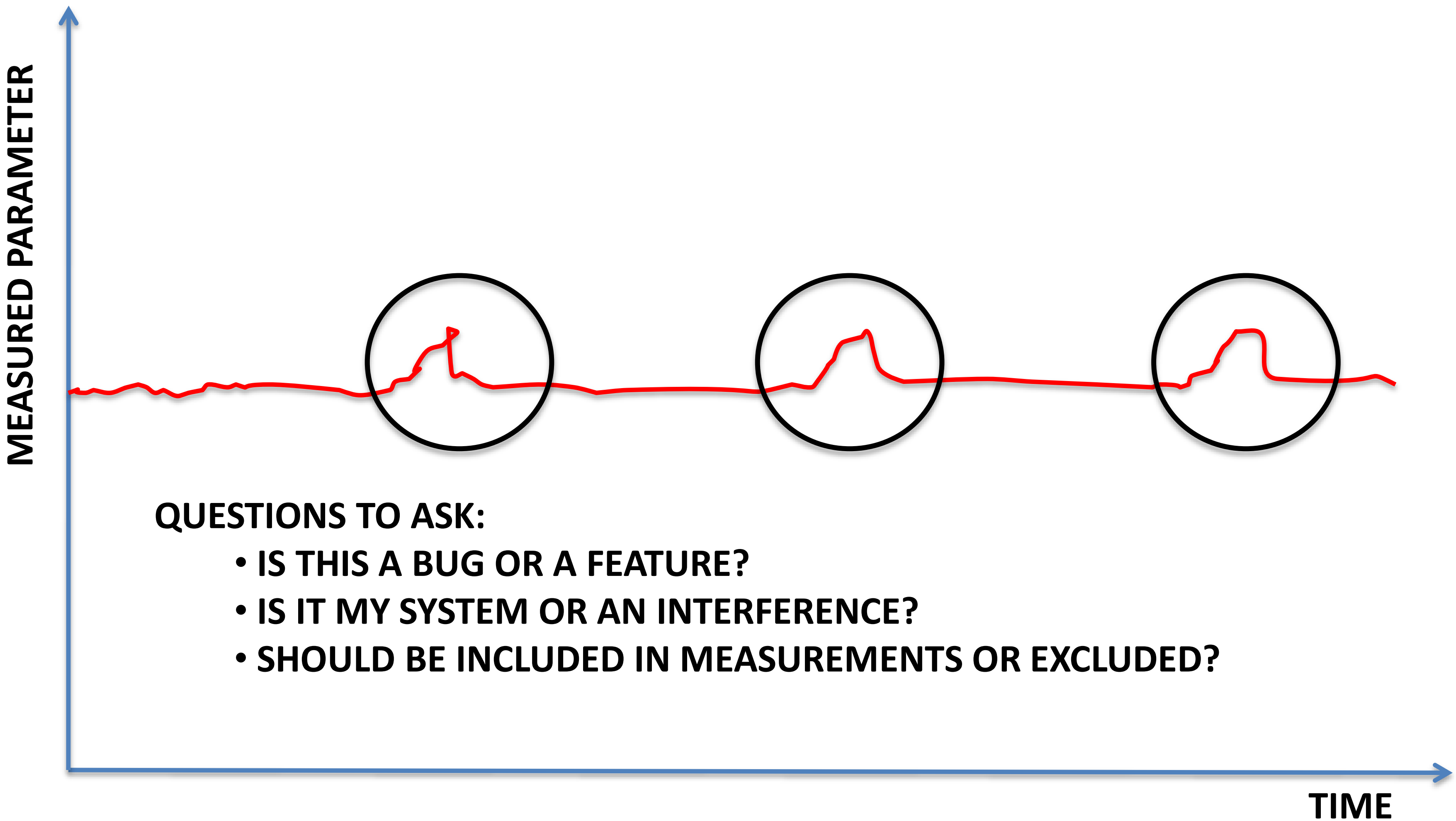
Warm up phase

- Warm up phase
  - Time until clients are all up, caches full (warm), data in main memory, etc.
  - Throughput lower than steady state throughput
  - Response time better than in steady state
- Detect by watching measured parameter changing with time
- Measure only in steady state

Cool down phase

- Cool down phase
  - Clients start finishing, resulting in less load in the system
  - Throughput is lower than in steady state
  - Response time better than in steady state
- Detect by observing when measured parameter suddenly changes behavior
- Stop measuring when clients no longer generate a steady load

Patterns to watch for - glitches

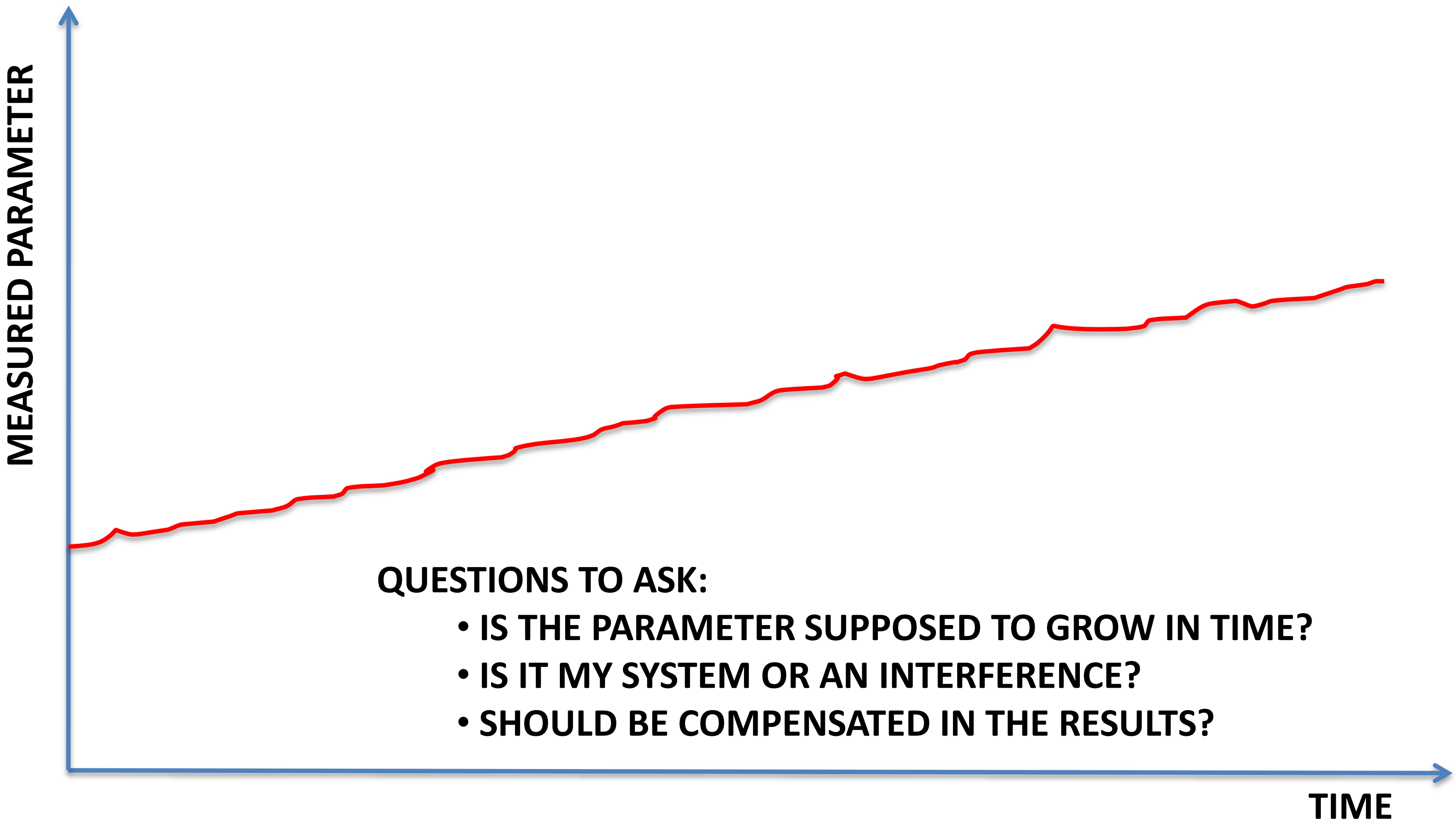


QUESTIONS TO ASK:

- **IS THIS A BUG OR A FEATURE?**
- **IS IT MY SYSTEM OR AN INTERFERENCE?**
- **SHOULD BE INCLUDED IN MEASUREMENTS OR EXCLUDED?**

ASSUME STEADY STATE MEASUREMENTS

Patterns to watch for - trends

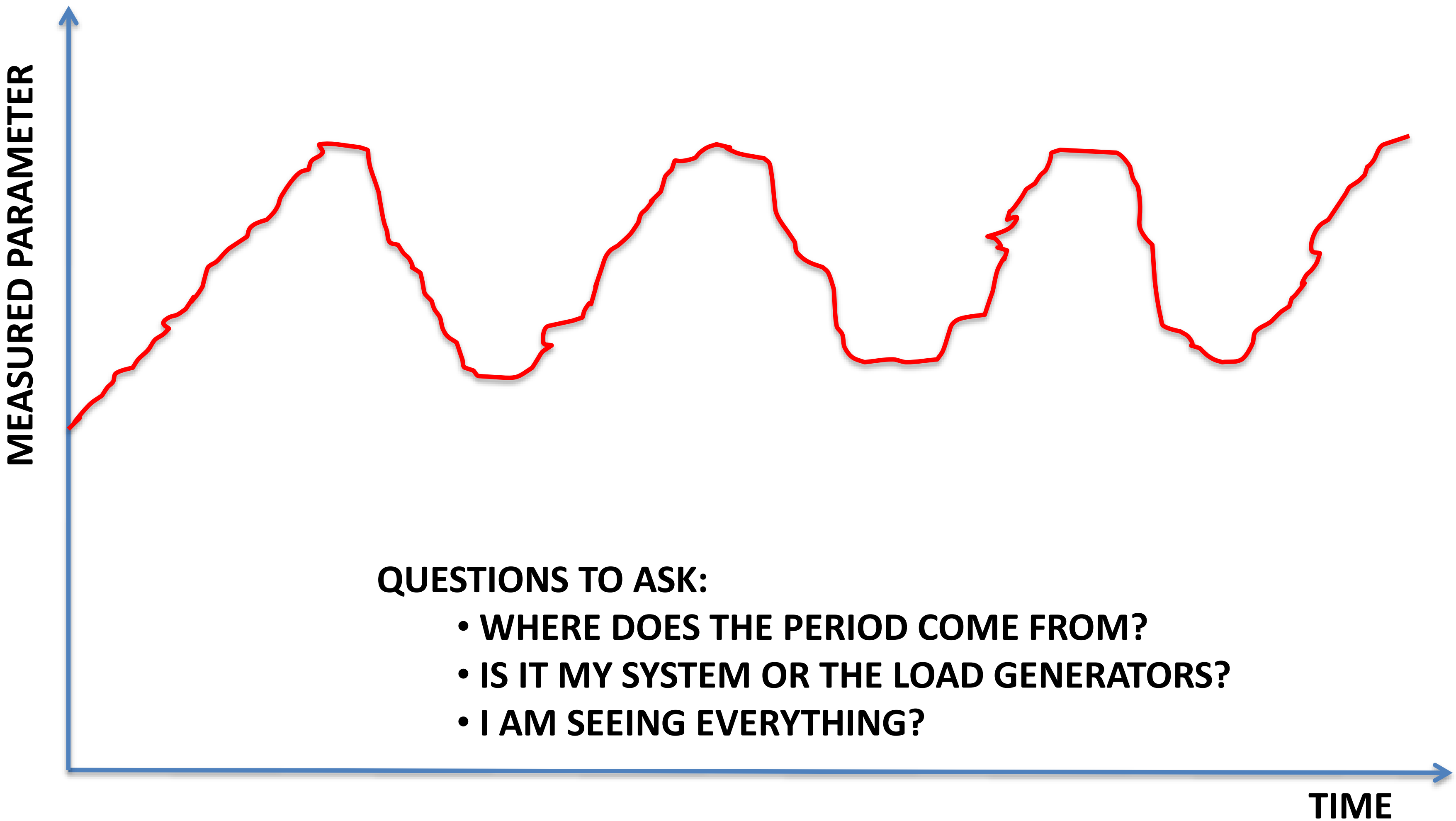


QUESTIONS TO ASK:

- **IS THE PARAMETER SUPPOSED TO GROW IN TIME?**
- **IS IT MY SYSTEM OR AN INTERFERENCE?**
- **SHOULD BE COMPENSATED IN THE RESULTS?**

ASSUME STEADY STATE MEASUREMENTS

Patterns to watch for - periodic



QUESTIONS TO ASK:

- **WHERE DOES THE PERIOD COME FROM?**
- **IS IT MY SYSTEM OR THE LOAD GENERATORS?**
- **I AM SEEING EVERYTHING?**

ASSUME STEADY STATE MEASUREMENTS

Why are these pattern relevant?

- Too few measurements and too short experiments are meaningless
  - May not capture system behavior
  - May not show pathological behavior
  - May not reflect real values
- Statistics are a way to address some of these issues by providing more information from the data and a better idea of the system behavior
  - but applying statistics to the wrong data will not help!

Data distributions

What are we measuring?

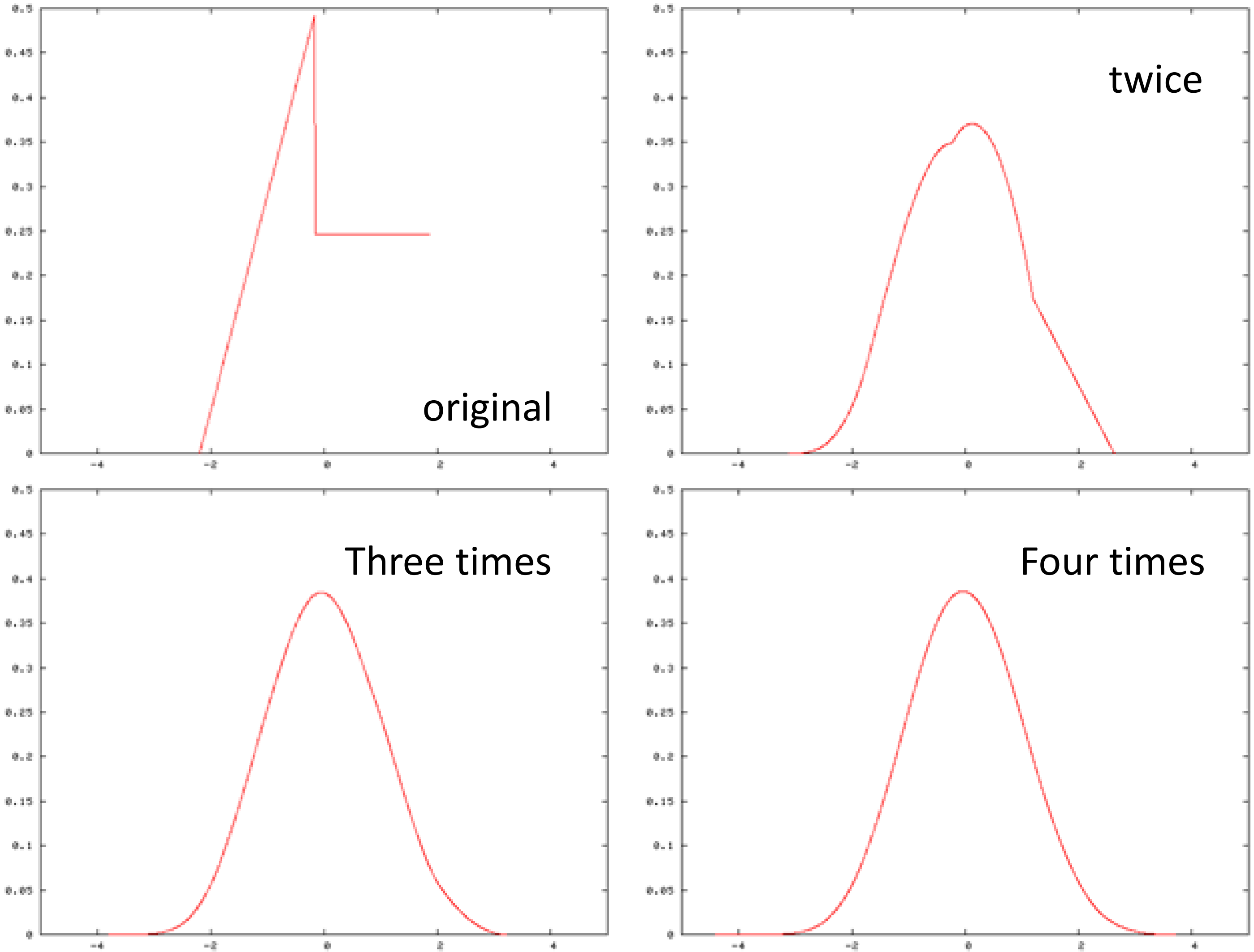
- When measuring, we are trying to estimate the value of a given parameter
- The value of the parameter is often determined by a complex combination of many effects and is typically not a constant
- Thus, the parameter we are trying to measure can be seen as a RANDOM VARIABLE
- The assumption is that this random variable has a NORMAL (GAUSSIAN) DISTRIBUTION

Central limit theorem

- Let X_1, X_2, X_3, … X_n be a sequence of independently and identically distributed random variables with finite values of
  - Expectation (μ)
  - Variance (σ^2)

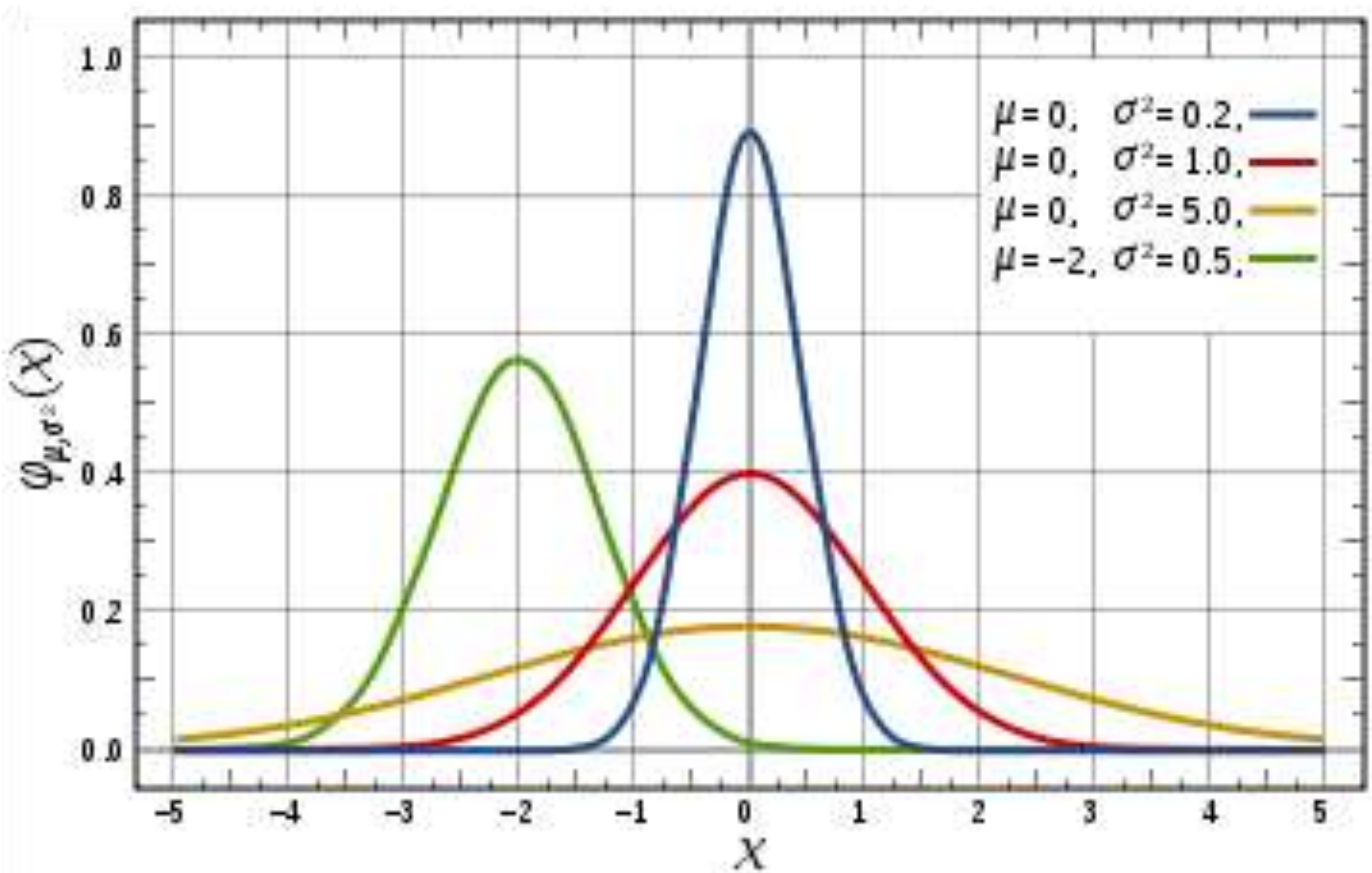
as the sample size n increases, the distribution of the sample average of the n random variables approaches the normal distribution with a mean μ and variance σ^2/n regardless of the shape of the original distribution.

How does it work?



http://en.wikipedia.org/wiki/File:Central_limit_thm.png

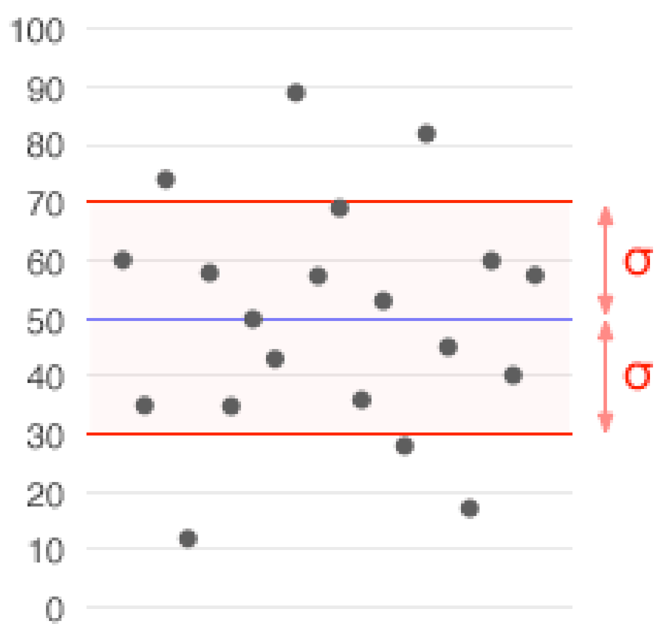
Normal or Gaussian distribution



http://en.wikipedia.org/wiki/Normal_distribution

Meaning of standard deviation

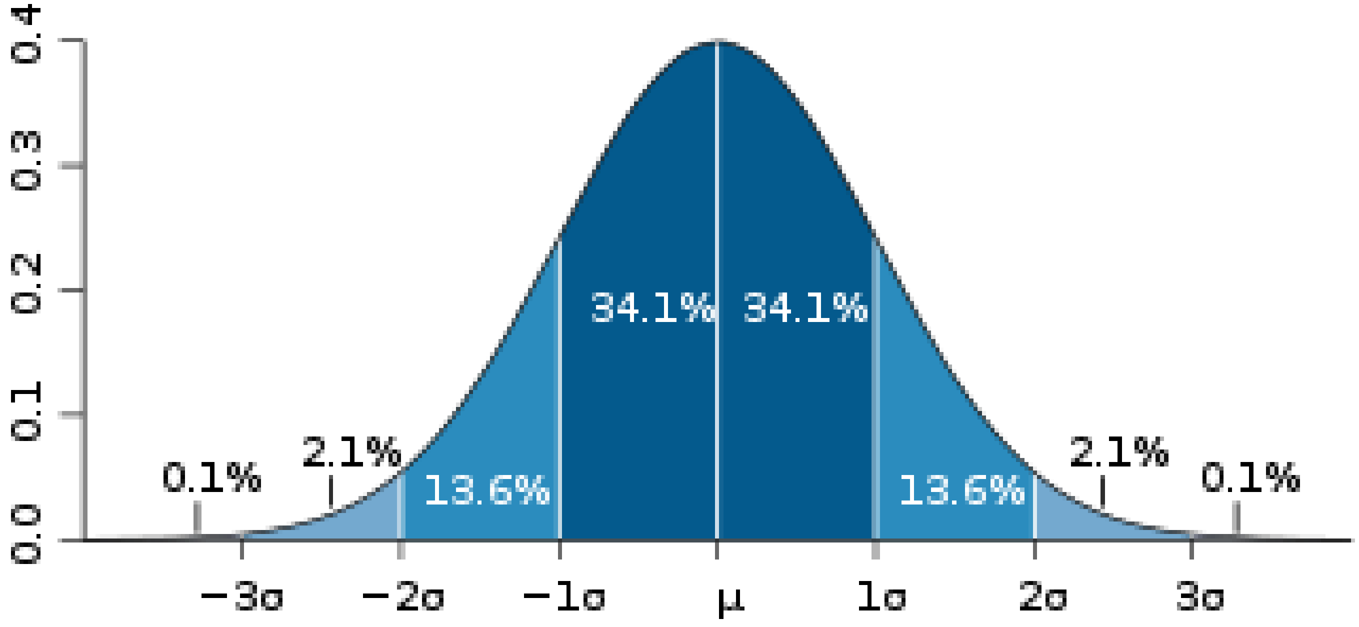
- The standard deviation gives an idea of how close are the measurements to the mean
  - important in defining service guarantees
  - important to understand real behavior



Mean is 50 but standard deviation is 20, indicating that there is a large variation in the value of the parameter

Mean of a sample

- To interpret a given measurement, we need to provide complete information
  - The mean
  - The standard deviation around the mean



http://en.wikipedia.org/wiki/Standard_deviation

Mean and standard deviation

- The standard deviation defines margins around the mean:
  - 64% of the values are within $\mu \pm \sigma$
  - 95% of the values are within $\mu \pm 2\sigma$
  - 99.7% of the values are within $\mu \pm 3\sigma$
- For a real system is very important to understand what happens when the values go beyond those margins (delays, overload, thrashing, system crash, etc.)

Calculating the standard deviation

- Mean and standard deviation:

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \qquad \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

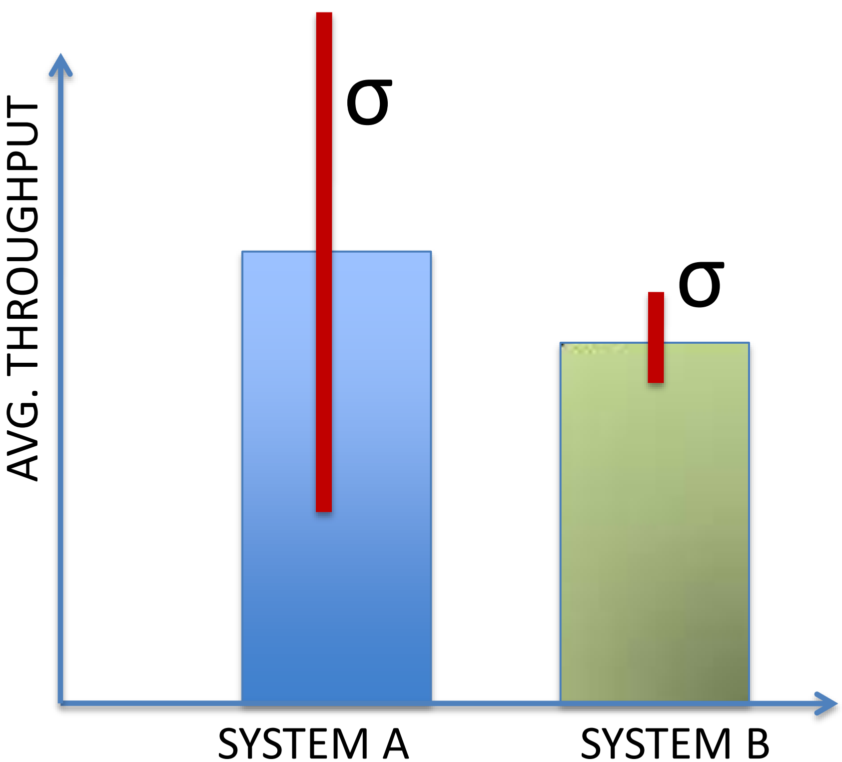
- In practice, use:

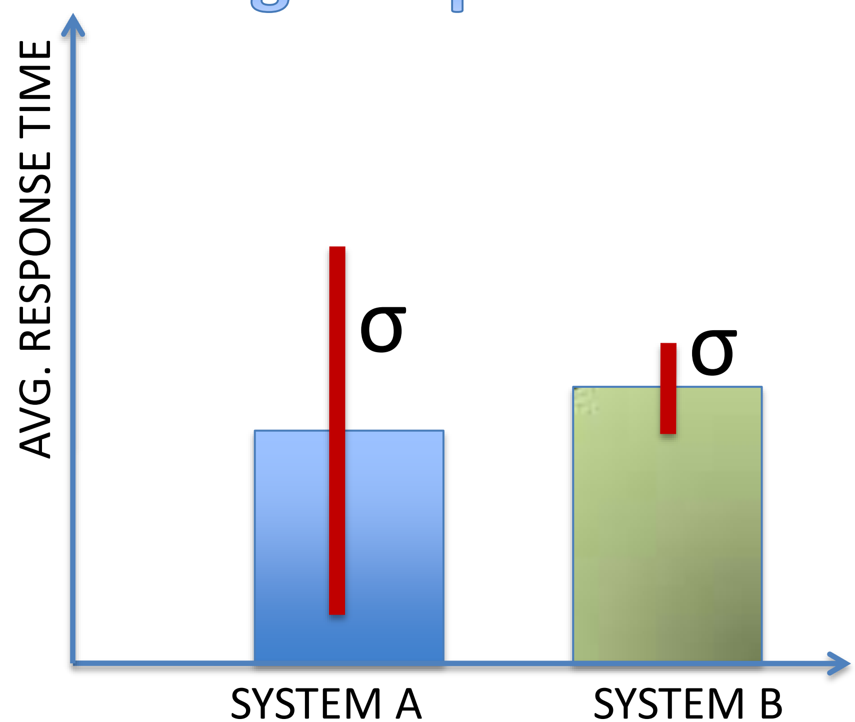
$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Comparisons

- What is better?

Deterministic behavior is often more important than good performance





In practice

- In many systems, the standard deviation is almost more important than the mean:
  - 90% of the queries need to be answered in less than X seconds
  - No web request can take longer than 5 seconds
  - Changes have to be propagated in less than 10 seconds
  - Guaranteed bandwidth higher than X 90% of the time
- Achieving determinism is often done at the cost of performance

Confidence intervals

Background

- When measuring in software system, we typically do know neither the value of the parameter we are measuring (μ) nor its standard deviation (σ)
- Instead, we work with mean of the sample $\overline{x}$ and the estimated standard deviation (s)
  - the result is no longer a normal distribution but a t-distribution
  - The t-distribution depends on n, the amount of samples
  - For large n, the t.distribution tends to a normal distribution

Confidence interval

- Since typically we are not measuring an absolute value (unlike in the natural sciences), the notion of confidence interval is particularly useful in computer science
- A confidence interval is a range (typically around the mean) where we can say that if we repeat the experiment 100 times, the value observed will be within the confidence interval m times (e.g., m=95, leading to a 95% confidence interval)

Calculation

- The confidence interval is calculated as follows:

$$CI = \bar{x} \pm t \cdot \frac{s}{\sqrt{n}}$$

Where s is the sample standard deviation, n the number of samples and t the critical value of the t-distribution

T in a table

Look up the value for the desired confidence interval and (n-1)

One Sided	75%	80%	85%	90%	95%	97.5%	99%	99.5%	99.75%	99.9%	99.95%
Two Sided	50%	60%	70%	80%	90%	95%	98%	99%	99.5%	99.8%	99.9%
1	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	127.3	318.3	636.6
2	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	14.09	22.33	31.60
3	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	7.453	10.21	12.92
4	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	5.598	7.173	8.610
5	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	4.773	5.893	6.869
6	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	4.317	5.208	5.959
7	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.029	4.785	5.408
8	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	3.833	4.501	5.041
9	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	3.690	4.297	4.781

Some observations

- For a fixed n
  - Increasing the confidence (100%(1-α)) implies to extend the confidence interval
- To reduce the confidence interval
  - we decrease the confidence or,
  - we increase the number of examples
- For experiments, fix a target (typically 95% confidence in a 5-10% interval around the mean) and repeat the experiments until the level of confidence is reached –if ever …

Example

- Mean = 122
- s = 9
- n = 30
- t (two sided, 29, 95%) = 2.045
- CI = 122 ± (2.045 · 9/30^{-2})
- In 95 out of 100 runs, the mean will be between 119 and 125

Putting it all together

Look at all the data

- Make sure you are looking at the complete picture of the experiment and your measurements do not include side effects (warm up, cool down, repetition effects)
- Once you are sure you have identified the valid data and that it looks reasonable, then apply statistics to it

Standard deviation

- All measurements and graphs have to be accompanied by the standard deviation, otherwise they are meaningless
  - Provides an idea of the precision
  - Provides an idea of what will happen in practice
  - Provides an idea of how predictable performance is
- Repeat the experiments until you get a reasonable standard deviation (enough values are close enough to the mean)

How long to run?

- Until you reach a reasonable confidence level that the value lies within a reasonable margin of the mean
- Confidence intervals are the way to quantify how often the reported result is going to be observed in practice
  - "we repeated the experiments until we reached a 95% level confidence for an interval 5% around the mean"

Advice

- It is a good idea to run a long experiment to make sure you have seen all possible behavior of the system:
  - Glitches only every 3 hours
  - Memory leaks after 1 M transactions
- In reality, tests have to resemble how the system will be used in practice

Designing an experiment

Experiments, but which ones?

- What does it mean to design an experiment?
- Performance is affected by a large number of factors
  - Workloads
  - Systems
  - Knobs
- We are interested in:
  - Which ones are the most important?
  - Which ones are related?
- Goal: get the most information with least effort (minimum number of experiments)

Definitions

- A *response variable* is the outcome of an experiment - typically the measured performance of the system (e.g., throughput, response time)
- A *factor* is any variable that affects the response, and which has several alternatives (amount of memory, number of cores, data sizes)

- *Levels* are the values that a given factor can assume – the alternatives for a factor.
- *Primary factors* are those whose effects need to be quantified
- *Secondary factors* are those that impact performance but whose effect we are not interested in quantifying
- *Replications* are the number of times each experiment is to be repeated with particular levels for each factor.

An experiment

- An experimental design consists of:
  - the number of different experiments
  - the factor level combinations for each experiment
  - the number of replications of each experiment
- An experimental unit is any entity used for the experiment

Interaction

- Two factors interact if the effect of one depends on the level of the other.
- Interaction considerably complicates the business of interpreting experimental results

Non-Interacting

	A_1	A_2
B_1	3	5
B_2	6	8

Interacting

	A_1	A_2
B_1	3	5
B_2	6	9

Avoid mistakes

- Try to avoid the following:
  - Ignoring the variation due to experimental errors
  - Not controlling important parameters (secondary factors)
  - Not isolating the effects of different factors
  - Overly simple (and very inefficient designs)
  - Ignoring interactions between factors
  - Conducting too many experiments
    - Take it slowly!
    - Break up the project into steps

Exploring the space

- Given a number of factors, what to do?
- Bad idea:
  - Vary one factor at a time
  - Find best value, fix it
  - Repeat for each factor
- Why is this a bad idea?: too many experiments, will get stuck in local mimimum

2^k Factorial Designs

- Experimental technique to find the relative weight of different factors
  - Pick K factors
  - Pick two levels for each factor
  - Behavior of factors must be unidirectional or monotonic in the range explored (!)

Example 2^2 Factorial Design

Observation: can update book examples by multiplying by 1000!

	Memory size	
Cache size (MB)	4GB	16GB
1	15	45
2	25	75

Define variables x_A and x_b to represent levels for each factor:

$$x_A = \begin{cases} -1 & \text{if 4 GB main memory;} \\ 1 & \text{if 16 GB main memory.} \end{cases}$$

$$x_B = \begin{cases} -1 & \text{if 1 MB cache,} \\ 1 & \text{if 2 MB cache.} \end{cases}$$

Solving the model

A useful fiction: non-linear regression model for performance:

$$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B$$

This means we can write:

$$\begin{aligned}
15 &= q_0 - q_A - q_B + q_{AB} \\
45 &= q_0 + q_A - q_B - q_{AB} \\
25 &= q_0 - q_A + q_B - q_{AB} \\
75 &= q_0 + q_A + q_B + q_{AB}
\end{aligned}$$

Relative weights on response variable

Solving:

$$y = 40 + 20x_A + 10x_B + 5x_Ax_B$$

What does this mean?

- Mean performance is 40
- Effect of memory is 20
- Effect of cache is 10
- Interaction between the two accounts for 5

In the form of a table

The same calculation can be done using a sign table:

I	A	B	AB	y
1	-1	-1	1	15
1	1	-1	-1	45
1	-1	1	-1	25
1	1	1	1	75
160	80	40	20	Total
40	20	10	5	Total/4

Repetitions and errors

- Look in the book
  - How to allocate variation
  - How to consider repetitions of the experiments to look for errors

- For the milestone and analysis, please use repetitions to get meaningful results.