CMPT 473
Software Quality Assurance

Performance

Nick Sumner - Fall 2014

Performance

- Performance is often considered a key nonfunctional criterion, but it is often poorly analyzed.

Performance

- Performance is often considered a key nonfunctional criterion, but it is often poorly analyzed.
- Need to keep several several issues in mind:

Performance

- Performance is often considered a key nonfunctional criterion, but it is often poorly analyzed.
- Need to keep several several issues in mind:
  - What facets of performance matter?

Performance

- Performance is often considered a key nonfunctional criterion, but it is often poorly analyzed.
- Need to keep several several issues in mind:
  - What facets of performance matter?
  - What component/granularity matters?

Performance

- Performance is often considered a key nonfunctional criterion, but it is often poorly analyzed.
- Need to keep several several issues in mind:
  - What facets of performance matter?
  - What component/granularity matters?
  - How can we appropriately measure/ensure them?

Performance

- Performance is often considered a key nonfunctional criterion, but it is often poorly analyzed.
- Need to keep several several issues in mind:
  - What facets of performance matter?
  - What component/granularity matters?
  - How can we appropriately measure/ensure them?
  - How should we report/present the results?

Performance

- Performance is often considered a key nonfunctional criterion, but it is often poorly analyzed.
- Need to keep several several issues in mind:
  - What facets of performance matter?
  - What component/granularity matters?
  - How can we appropriately measure/ensure them?
  - How should we report/present the results?
  - How can this fit into the overall QA process?

Facets of Performance

- What do we mean by performance?

Facets of Performance

- What do we mean by performance?
  - Speed/Runtime
    - Total running time required

Facets of Performance

- What do we mean by performance?
  - Speed/Runtime
    - Total running time required
  - Throughput
    - Pages/Transactions per second, bytes per second

Facets of Performance

- What do we mean by performance?
  - Speed/Runtime
    - Total running time required
  - Throughput
    - Pages/Transactions per second, bytes per second
  - Responsiveness
    - UI response time, server response time at peak load

Facets of Performance

- What do we mean by performance?
  - Speed/Runtime
    - Total running time required
  - Throughput
    - Pages/Transactions per second, bytes per second
  - Responsiveness
    - UI response time, server response time at peak load
  - Memory Consumption
    - Peak memory consumption
  - ...

Facets of Performance

- All of these (and more) are valid concerns

Facets of Performance

- All of these (and more) are valid concerns
- You must think critically about what aspects are most important to
  - End users?

Facets of Performance

- All of these (and more) are valid concerns
- You must think critically about what aspects are most important to
  - End users?
  - Operators?

Coarseness of Performance

- Performance may matter at different levels

Coarseness of Performance

- Performance may matter at different levels
  - Whole system

Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components

Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components
  - Instruction level architectural level

Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components
  - Instruction level architectural level
- May be consider performance at many different levels, but they may not be independent!

Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components
  - Instruction level architectural level
- May be consider performance at many different levels, but they may not be independent!

How are the different levels related?

Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components
  - Instruction level architectural level
- May be consider performance at many different levels, but they may not be independent!
  - Whole system & critical components are monitored for performance.

Coarseness of Performance

- Performance may matter at different levels
  - Whole system
  - Critical components
  - Instruction level architectural level
- May be consider performance at many different levels, but they may not be independent!
  - Whole system & critical components are monitored for performance.
  - Instruction and architectural level issues are examined only on demand when other problems are noted.

Coarseness of Performance

- System level issues?

Coarseness of Performance

- System level issues?
  - Design
  - Algorithms

Coarseness of Performance

- System level issues?
  - Design
  - Algorithms
- Instruction/Architecture level issues?

Coarseness of Performance

- System level issues?
  - Design
  - Algorithms
- Instruction/Architecture level issues?
  - Cache misses
  - Branch misprediction
  - Underutilization of resources (SIMD, multicore, …)

Coarseness of Performance

- System level issues?
  - Design
  - Algorithms
- Instruction/Architecture level issues?
  - Cache misses
  - Branch misprediction
  - Underutilization of resources (SIMD, multicore, …)
- Use profilers to gain information at a certain level
  - perf, gprof, cachegrind, oprofile, java mission control
  - Purify, VS Profiling Tools

Measurement

- So how can we measure these?
  - Idea: Run the test suite and see how long it takes

Is this reasonable? Why/why not?

Measurement

- So how can we measure these?
  - Idea: Run the test suite and see how long it takes
    - How well does this capture system level performance?
    - “ “ low level performance?

Measurement

- So how can we measure these?
  - Idea: Run the test suite and see how long it takes
    - How well does this capture system level performance?
    - “ “ low level performance?
- A *functionality* based test suite will not reflect the *performance* concerns!

Measurement

- So how can we measure these?
  - Idea: Run the test suite and see how long it takes
    - How well does this capture system level performance?
    - “ “ low level performance?
- A *functionality* based test suite will not reflect the *performance* concerns!
  - Need to design a new set of tests specifically targeting performance issues.

Measurement

- So how can we measure these?
  - Idea: Run the test suite and see how long it takes
    - How well does this capture system level performance?
    - “ “ low level performance?
- A *functionality* based test suite will not reflect the *performance* concerns!
  - Need to design a new set of tests specifically targeting performance issues.

How? What should the tests capture?

Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures

Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures
  - Common interaction patterns

Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures
  - Common interaction patterns
  - Hot paths, tight loops, frequently called functions, …

Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures
  - Common interaction patterns
  - Hot paths, tight loops, frequently called functions, …
- Step 2: Go back to step 1 and make sure it is realistic
  - People commonly use synthetic workloads for performance measurement and are misled by results.

Why is this an issue?

Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures
  - Common interaction patterns
  - Hot paths, tight loops, frequently called functions, …
- Step 2: Go back to step 1 and make sure it is realistic
  - People commonly use synthetic workloads for performance measurement and are misled by results.

Measurement

- Step 1: Identify the workload(s) of interest
  - Expected input structures
  - Common interaction patterns
  - Hot paths, tight loops, frequently called functions, …
- Step 2: Go back to step 1 and make sure it is realistic
  - People commonly use synthetic workloads for performance measurement and are misled by results.
    - Too simple
    - Doesn't reflect interaction complexities of real code
    - ...

Measurement

- Step 3: Run & measure your workload

Measurement

- Step 3: Run & measure your workload
  - Is this good enough?

```
startTime = getCurrentTimeInSeconds();
doWorkloadOfInterest();
endTime = getCurrentTimeInSeconds();
reportResult(endTime – startTime);
```

Measurement

- Step 3: Run & measure your workload
  - Is this good enough?

```
startTime = getCurrentTimeInSeconds();
doWorkloadOfInterest();
endTime = getCurrentTimeInSeconds();
reportResult(endTime – startTime);
```

  - What can go wrong?

Measurement

- Step 3: Run & measure your workload
  - Is this good enough?

```
startTime = getCurrentTimeInSeconds();
doWorkloadOfInterest();
endTime = getCurrentTimeInSeconds();
reportResult(endTime – startTime);
```

  - What can go wrong?
    - Timing granularity & units
    - Warm up time
    - Nondeterministic behavior
    - ...

Measurement

- Granularity & Units
  - Why is granularity a problem?

Measurement

- Granularity & Units
  - Why is granularity a problem?
  - What are alternatives to getCurrentTimeInSeconds?

Measurement

- Granularity & Units
  - Why is granularity a problem?
  - What are alternatives to getCurrentTimeInSeconds?
  - What if I want to predict performance on a different machine?

Measurement

- Granularity & Units
  - Why is granularity a problem?
  - What are alternatives to getCurrentTimeInSeconds?
  - What if I want to predict performance on a different machine?
    - Using *cycles* instead of wall clock time can be useful, but has its own limitations

Measurement

- Warm up time
  - Why is warm up time necessary *in general*?

Measurement

- Warm up time
  - Why is warm up time necessary *in general*?
  - Why is it especially problematic for systems like Java?

Measurement

- Warm up time
  - Why is warm up time necessary *in general*?
  - Why is it especially problematic for systems like Java?
  - How can we modify our example to facilitate this?

Measurement

- Warm up time
  - Why is warm up time necessary *in general*?
  - Why is it especially problematic for systems like Java?
  - How can we modify our example to facilitate this?

```
for (…) doWorkloadOfInterest();
startTime = getCurrentTimeInSeconds();
doWorkloadOfInterest();
endTime = getCurrentTimeInSeconds();
reportResult(endTime – startTime);
```

Measurement

- Nondeterministic behavior
  - Will `getCurrentTimeInSeconds();` always return the same number?

Why/why not?

Measurement

- Nondeterministic behavior
  - Will `getCurrentTimeInSeconds();` always return the same number?
  - So what reflects a *meaningful* result?
    - Hint: The Law of Large Numbers!

Measurement

- Nondeterministic behavior
  - Will `getCurrentTimeInSeconds();` always return the same number?
  - So what reflects a *meaningful* result?
    - Hint: The Law of Large Numbers!
- By running the same test many times, the arithmetic mean will converge on the expected value

Measurement

- Nondeterministic behavior
  - Will `getCurrentTimeInSeconds();` always return the same number?
  - So what reflects a *meaningful* result?
    - Hint: The Law of Large Numbers!
- By running the same test many times, the arithmetic mean will converge on the expected value
  - So have we solved the problem?

Measurement

- Nondeterministic behavior
  - Will `getCurrentTimeInSeconds();` always return the same number?
  - So what reflects a *meaningful* result?
    - Hint: The Law of Large Numbers!
- By running the same test many times, the arithmetic mean will converge on the expected value
  - So have we solved the problem?
  - When do we stop?

Measurement

- Nondeterministic behavior
  - Will `getCurrentTimeInSeconds();` always return the same number?
  - So what reflects a *meaningful* result?
    - Hint: The Law of Large Numbers!
- By running the same test many times, the arithmetic mean will converge on the expected value
  - So have we solved the problem?
  - When do we stop?

There's a bit more to deal with here. We'll revisit it.

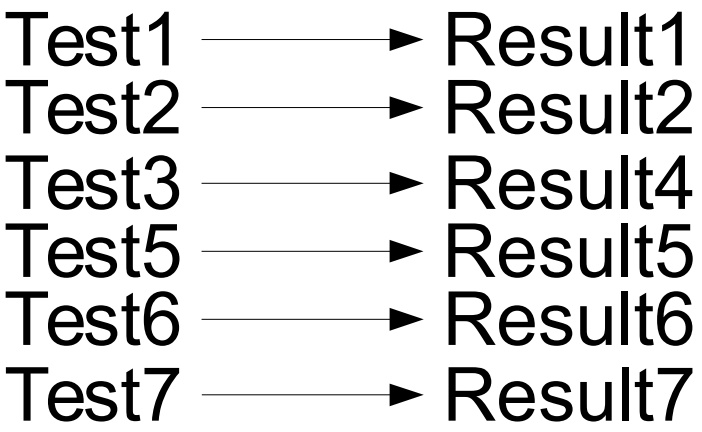
Measurement

- A revised (informal) approach:

```
for (…) doWorkloadOfInterest();
startTime = getCurrentTimeInNanos();
for (…) doWorkloadOfInterest();
endTime = getCurrentTimeInNanos();
reportResult(endTime – startTime);
```

Reporting

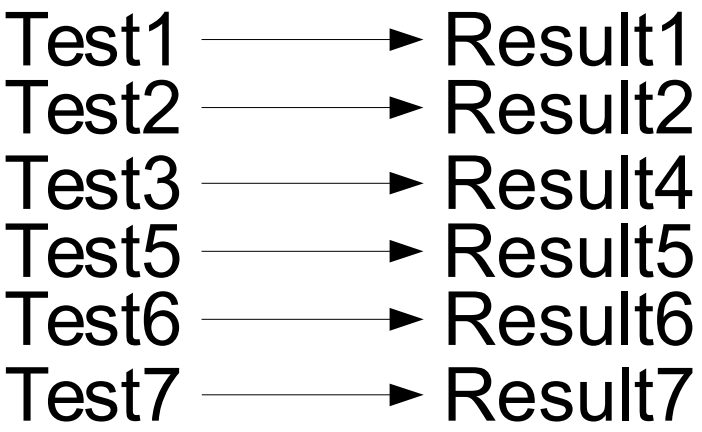
Congratulations! You have measured performance results of your application.



Are the results good or bad? How do you know?

Reporting

Congratulations! You have measured performance results of your application.



Are the results good or bad? How do you know?

The interpretation of results is only meaningful given ***context*** and appropriate ***presentation***.

Reporting

- Often, performance is tracked across versions

Old	New
Result1	Result1
Result2	Result2
Result4	Result4
Result5	Result5
Result6	Result6
Result7	Result7

Reporting

- Often, performance is tracked across versions

Old	New
Result1	Result1
Result2	Result2
Result4	Result4
Result5	Result5
Result6	Result6
Result7	Result7

- Given this, how can we tell if there is an improvement or a problem?

Reporting

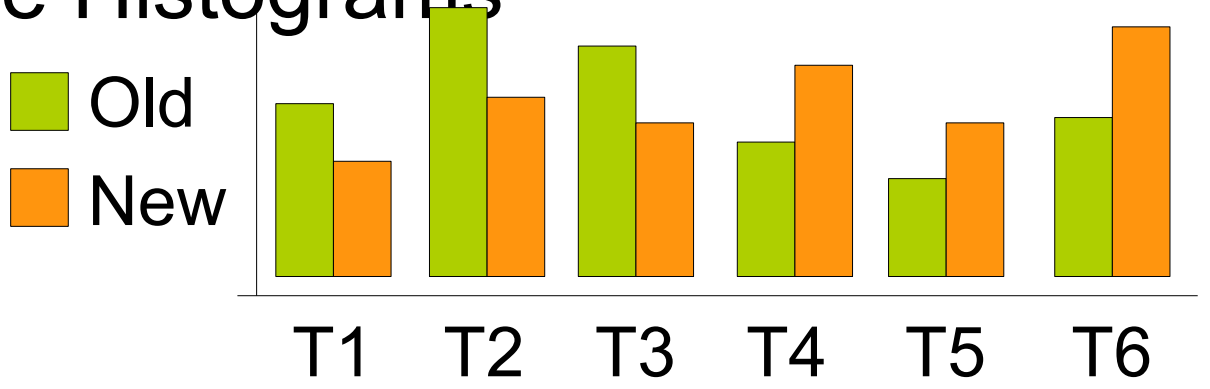
Strategies

- Sample Histograms

Reporting

Strategies

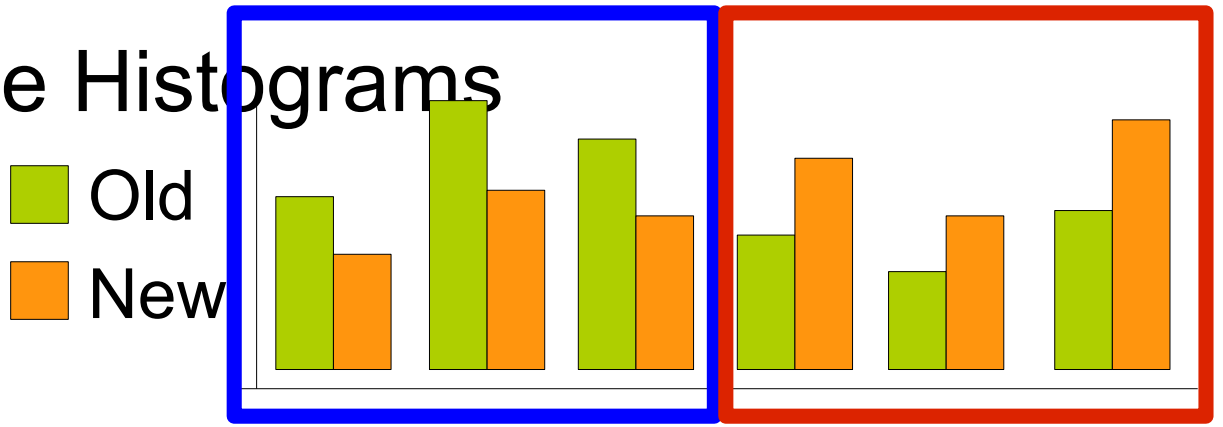
- Sample Histograms



Reporting

Strategies

- Sample Histograms

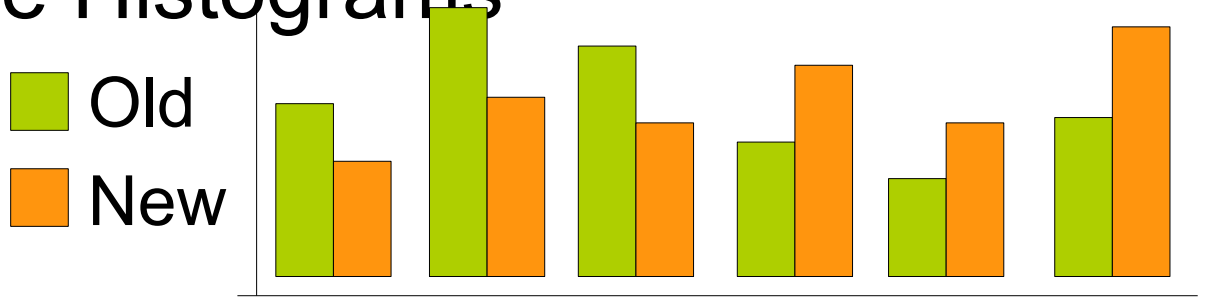


  - Both gains & losses can be noted

Reporting

Strategies

- Sample Histograms

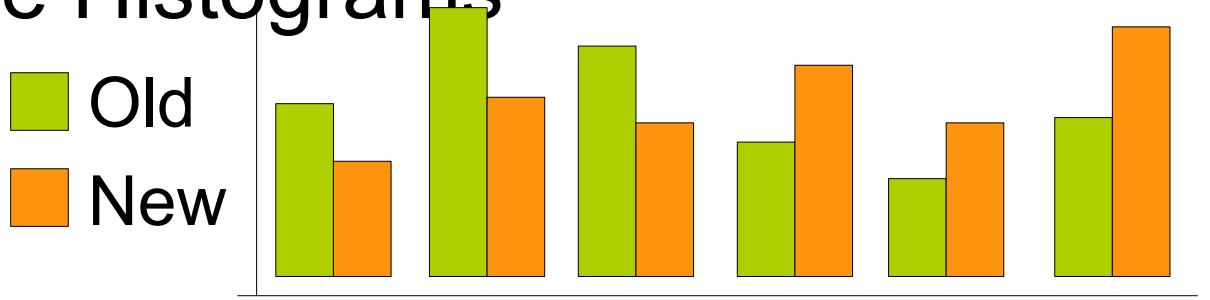


  - Both gains & losses can be noted
- Distribution of changes

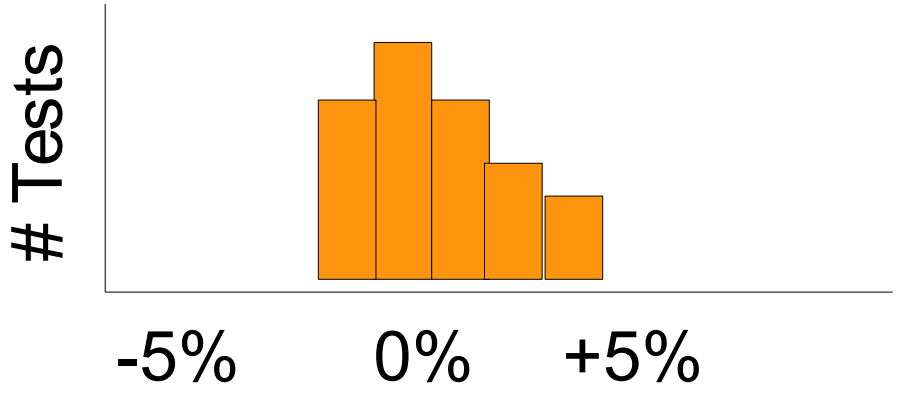
Reporting

Strategies

- Sample Histograms



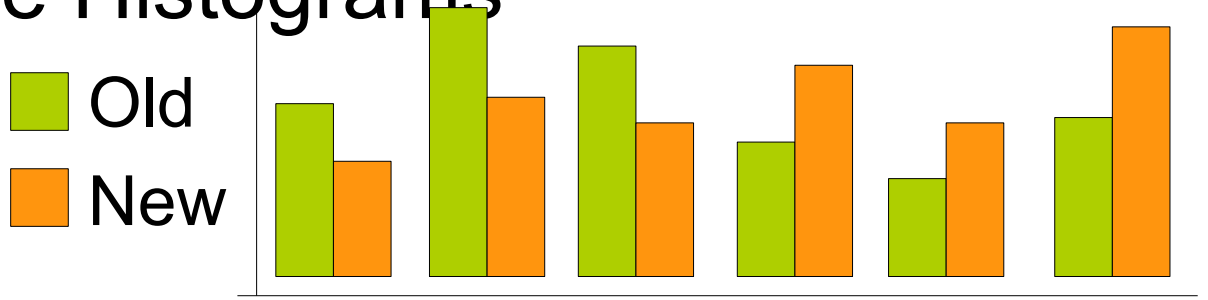
  - Both gains & losses can be noted
- Distribution of changes



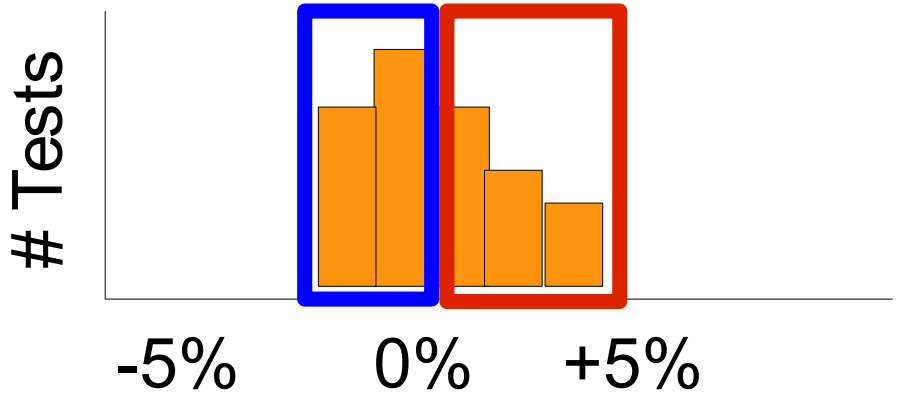
Reporting

Strategies

- Sample Histograms

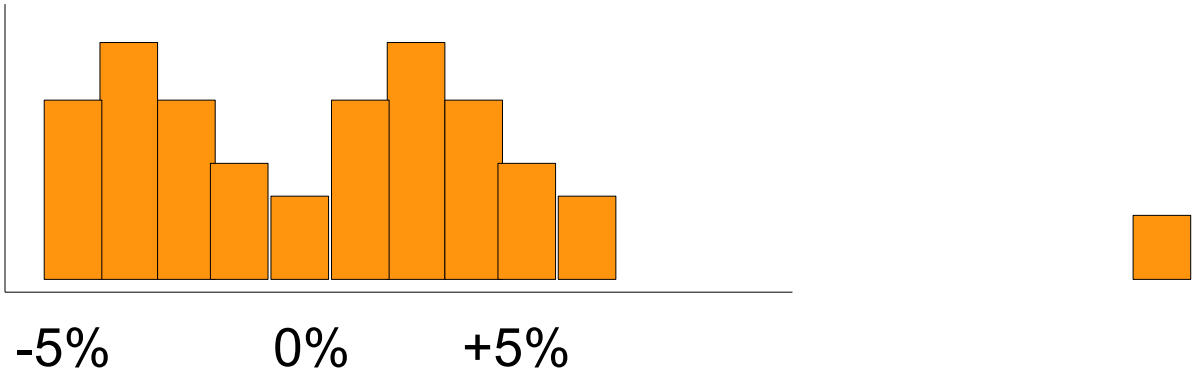


  - Both gains & losses can be noted
- Distribution of changes



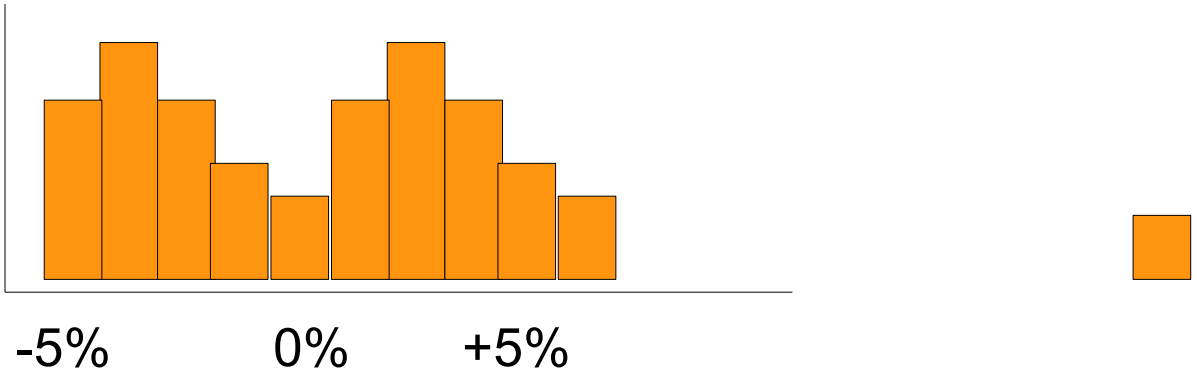
Reporting

- You might have to summarize the changes using a single number



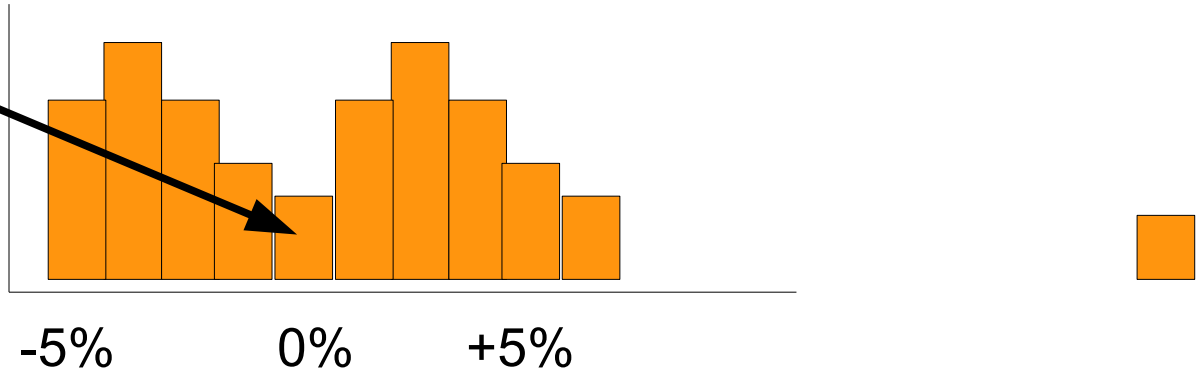
Reporting

- You might have to summarize the changes using a single number
  - Median?



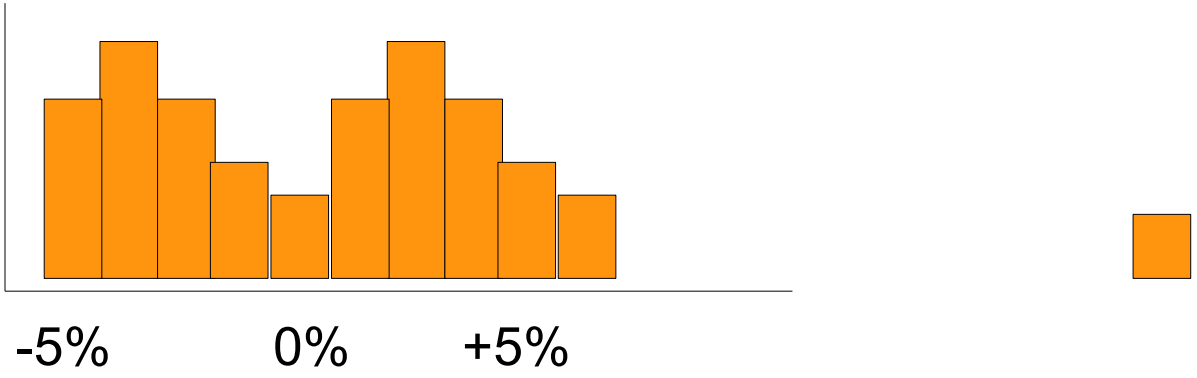
Reporting

- You might have to summarize the changes using a single number
  - Median?



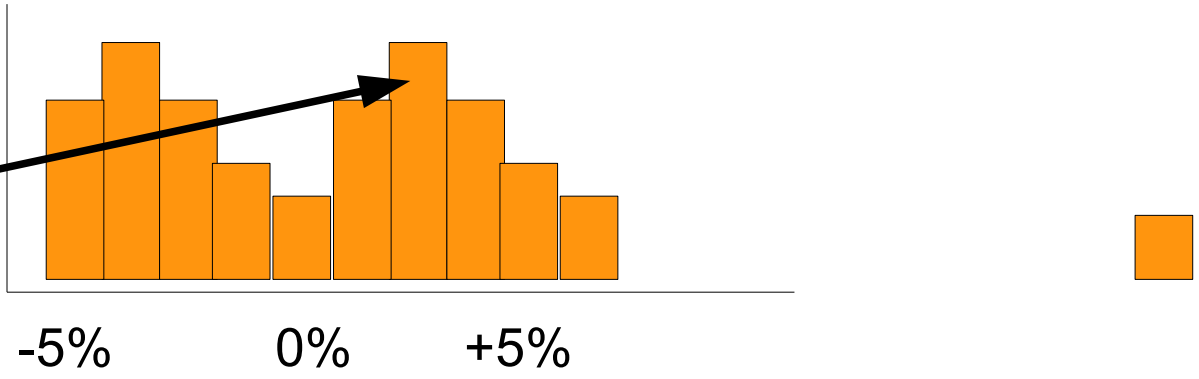
Reporting

- You might have to summarize the changes using a single number
  - Median?
  - Mode?



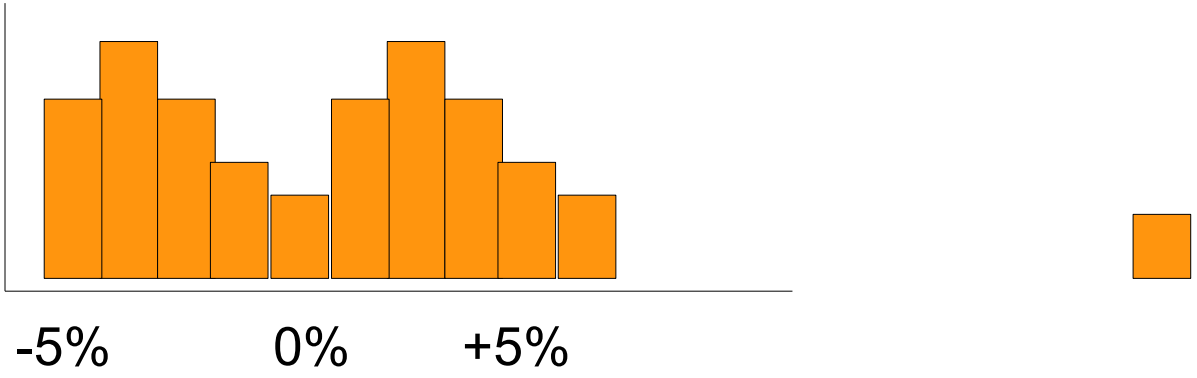
Reporting

- You might have to summarize the changes using a single number
  - Median?
  - Mode?



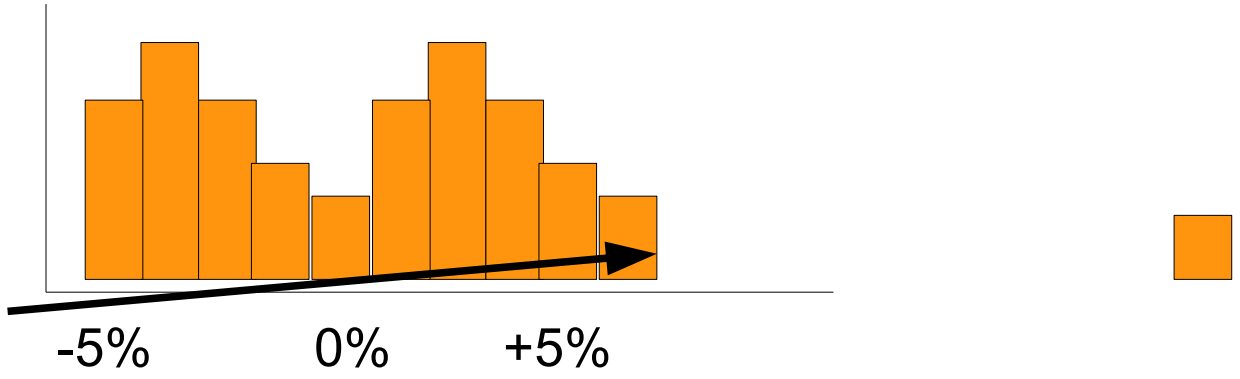
Reporting

- You might have to summarize the changes using a single number
  - Median?
  - Mode?
  - Mean?



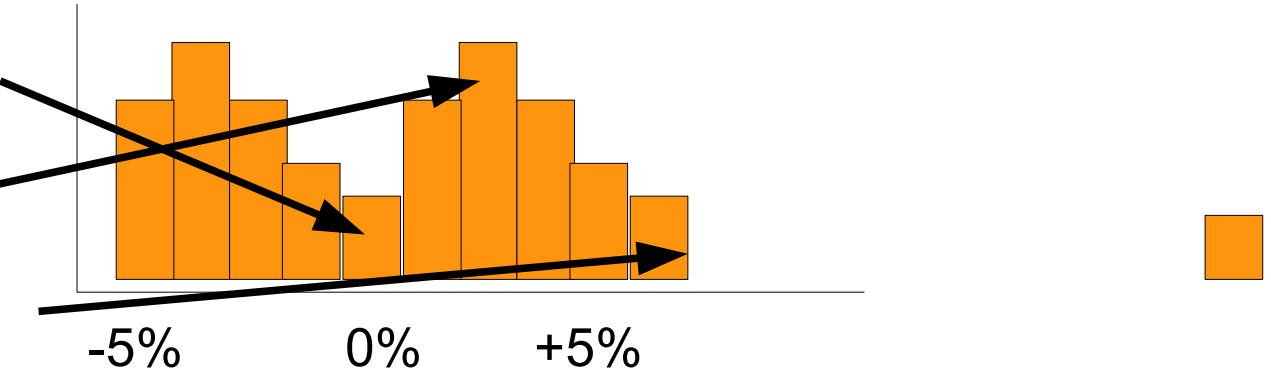
Reporting

- You might have to summarize the changes using a single number
  - Median?
  - Mode?
  - Mean?



Reporting

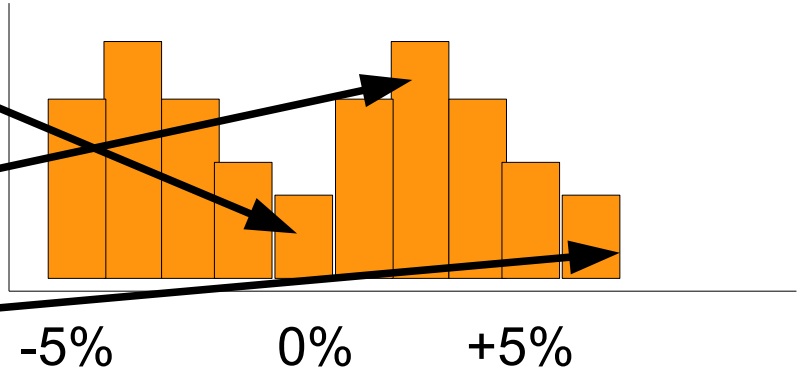
- You might have to summarize the changes using a single number
  - Median?
  - Mode?
  - Mean?



All of them can be misleading in some way!
Yet you'll still likely be expected to use them.

Reporting

- You might have to summarize the changes using a single number
  - Median?
  - Mode?
  - Mean?

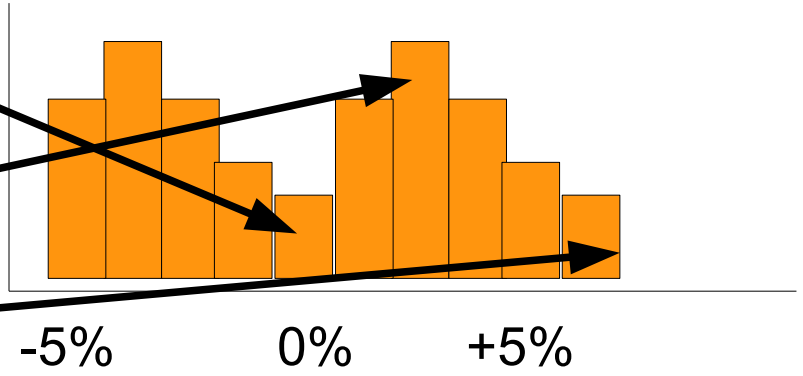




- But what do we even mean by... mean/average?

Reporting

- You might have to summarize the changes using a single number
  - Median?
  - Mode?
  - Mean?



- But what do we even mean by... mean/average?
  - There are actually many types of averages

Reporting

Averages of r_1, r_2, …, r_N

- Many ways to measure *expectation* or *tendency*

Reporting

Averages of $r_1, r_2, \ldots, r_N$

- Many ways to measure *expectation* or *tendency*
- Arithmetic Mean $\frac{1}{N}\sum_{i=1}^{N} r_i$

Reporting

Averages of $r_1, r_2, \ldots, r_N$

- Many ways to measure *expectation* or *tendency*
- Arithmetic Mean $\frac{1}{N}\sum_{i=1}^{N} r_i$
- Harmonic Mean $\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$

Reporting

Averages of $r_1, r_2, \ldots, r_N$

- Many ways to measure *expectation* or *tendency*
- Arithmetic Mean $\frac{1}{N}\sum_{i=1}^{N} r_i$
- Harmonic Mean $\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$
- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$

Reporting

Averages of $r_1, r_2, \ldots, r_N$

- Many ways to measure *expectation* or *tendency*
- Arithmetic Mean $\frac{1}{N}\sum_{i=1}^{N} r_i$
- Harmonic Mean $\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$
- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$

Each type means something different and has valid uses

Reporting

- Arithmetic Mean $\frac{1}{N}\sum_{i=1}^{N} r_i$
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing *sample means*

Reporting

- Arithmetic Mean $\frac{1}{N}\sum_{i=1}^{N} r_i$
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing sample means
  - e.g. Timing the same workload many times

Reporting

- Arithmetic Mean $\frac{1}{N}\sum_{i=1}^{N} r_i$
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing sample means
  - e.g. Timing the same workload many times

Handling Nondeterminism

```
for (x in 0 to 4)
  times[x] = doWorkloadOfInterest();
```

E(time) = arithmean(times)

Reporting

- Arithmetic Mean $\frac{1}{N}\sum_{i=1}^{N} r_i$
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing sample means
  - e.g. Timing the same workload many times
- Harmonic Mean
  - Good for reporting rates

$$\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$$

Reporting

- Arithmetic Mean $\frac{1}{N}\sum_{i=1}^{N} r_i$
  - Good for reporting averages of numbers that mean the same thing
  - Used for computing sample means
  - e.g. Timing the same workload many times
- Harmonic Mean $\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$
  - Good for reporting rates
  - e.g. Required throughput for a set of tasks

Reporting

Given tasks t1, t2, & t3 serving 40 pages each:

thoughput(t1) = 10 pages/sec

thoughput(t2) = 20 pages/sec

thoughput(t3) = 20 pages/sec

What is the average throughput? What should it mean?

- Harmonic Mean
  - Good for reporting rates
  - e.g. Required throughput for a set of tasks

$$\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$$

Reporting

Given tasks t1, t2, & t3 serving 40 pages each:

thoughput(t1) = 10 pages/sec

thoughput(t2) = 20 pages/sec

thoughput(t3) = 20 pages/sec

What is the average throughput? What should it mean?

Arithmetic = 16.7 p/s

- Harmonic Mean
  - Good for reporting rates
  - e.g. Required throughput for a set of tasks

$$\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$$

Reporting

Given tasks t1, t2, & t3 serving 40 pages each:

thoughput(t1) = 10 pages/sec

thoughput(t2) = 20 pages/sec

thoughput(t3) = 20 pages/sec

What is the average throughput? What should it mean?

Arithmetic = 16.7 p/s Harmonic = 15 p/s

- Harmonic Mean
  - Good for reporting rates
  - e.g. Required throughput for a set of tasks

$$\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$$

Reporting

Given tasks t1, t2, & t3 serving 40 pages each:

thoughput(t1) = 10 pages/sec

thoughput(t2) = 20 pages/sec

thoughput(t3) = 20 pages/sec

What is the average throughput? What should it mean?

Arithmetic = 16.7 p/s

120/16.7 = 7.2

Harmonic = 15 p/s

120/15 = 8

- Harmonic Mean
  - Good for reporting rates
  - e.g. Required throughput for a set of tasks

$$\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$$

Reporting

Given tasks t1, t2, & t3 serving 40 pages each:

thoughput(t1) = 10 pages/sec
thoughput(t2) = 20 pages/sec
thoughput(t3) = 20 pages/sec

What is the average throughput? What should it mean?

Arithmetic = 16.7 p/s
120/16.7 = 7.2

Harmonic = 15 p/s
120/15 = 8

$$\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$$

- Harmonic Mean
  - Good for reporting
  - e.g. Required throughput for a set of tasks

Identifies the constant rate required for the same time

Reporting

Given tasks t1, t2, & t3 serving 40 pages each:

thoughput(t1) = 10 pages/sec
thoughput(t2) = 20 pages/sec
thoughput(t3) = 20 pages/sec

What is the average throughput? What should it mean?

Arithmetic = 16.7 p/s
120/16.7 = 7.2

Harmonic = 15 p/s
120/15 = 8

$$\frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$$

- Harmonic Mean
  - Good for reporting
  - e.g. Required throughput for a set of tasks

Identifies the constant rate required for the same time

CAVEAT: If the size of each workload changes, a weighted harmonic mean is required!

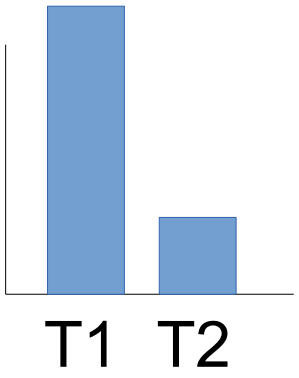
Reporting

- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks

Any idea why it may be useful here?
(A bit of a though experiment)

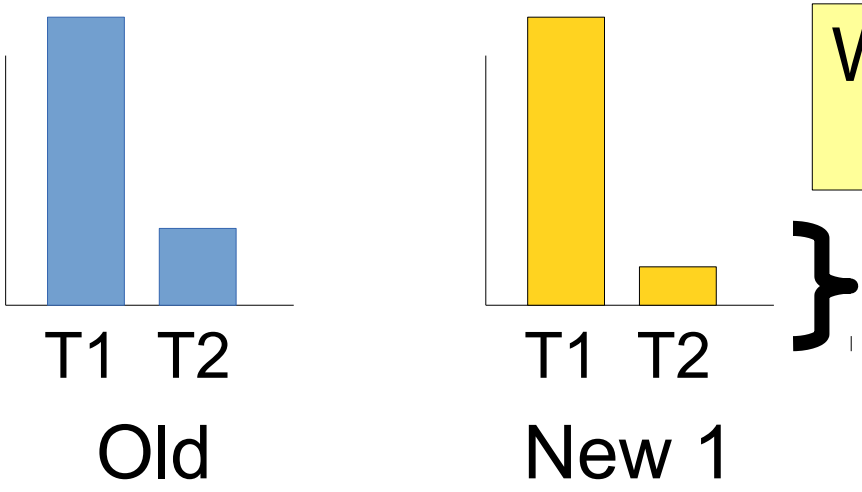
Reporting

- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks



Reporting

- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks

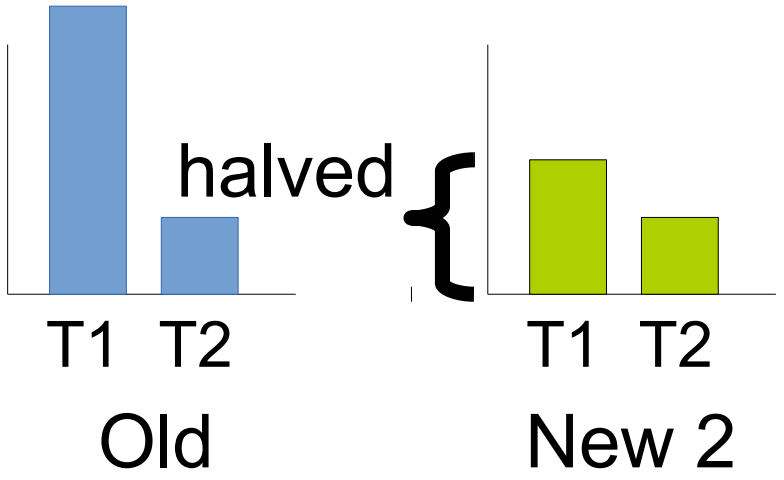


What happens to the arithmetic mean?

} halved

Reporting

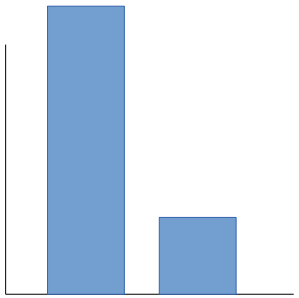
- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks



What happens to the arithmetic mean?

Reporting

- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks

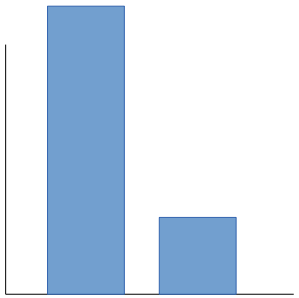


Old

The (non) change to T1 dominates any behavior for T2!

Reporting

- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks



Old

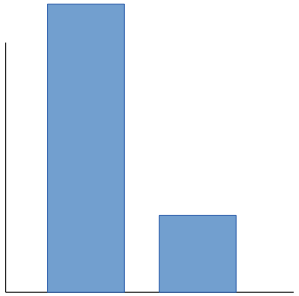
Geometric:

$\sqrt{r_1 \times r_2}$

Old

Reporting

- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks



Old

Geometric:

$\sqrt{r_1 \times r_2}$

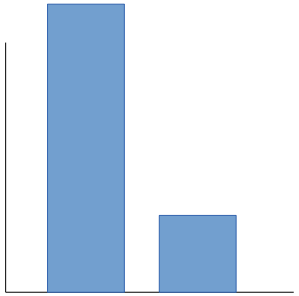
Old

$\sqrt{r_1 \times (\frac{1}{2} r_2)}$

New 1

Reporting

- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks



Old

Geometric:

$\sqrt{r_1 \times r_2}$

Old

$\sqrt{r_1 \times (\frac{1}{2} r_2)}$

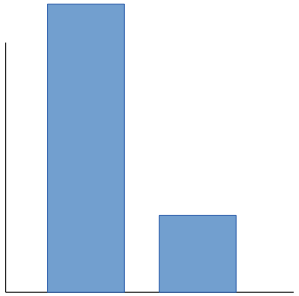
New 1

$\sqrt{(\frac{1}{2} r_1) \times r_2}$

New 2

Reporting

- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks



Old

Geometric:

$\sqrt{r_1 \times r_2}$

Old

$\sqrt{r_1 \times (\frac{1}{2} r_2)}$

New 1

$= \sqrt{\frac{1}{2} \times r_1 \times r_2} = \sqrt{(\frac{1}{2} r_1) \times r_2}$

New 2

Reporting

- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks
  - A 10% difference in any benchmark affects the final value the same way

Reporting

- Geometric Mean $\sqrt[N]{\prod_{i=1}^{N} r_i}$
  - Good for reporting results that mean different things
  - e.g. Timing results across *many different* benchmarks
  - A 10% difference in any benchmark affects the final value the same way

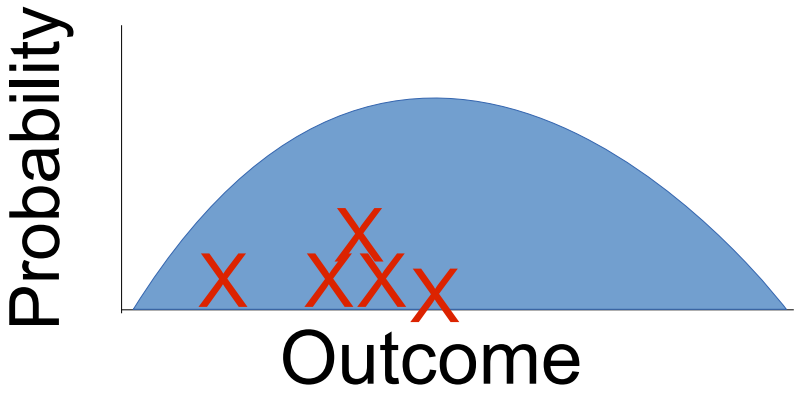
Note: It doesn't have an intuitive meaning!
It merely provides a balanced *score* of performance.

Reporting

- Nondeterminism
  - But all of these assume that the numbers across benchmarks are valid/deterministic
  - What does nondeterminism do?

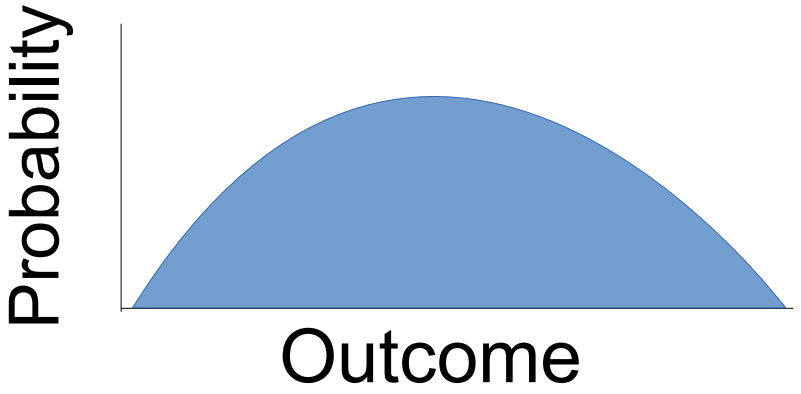
Reporting

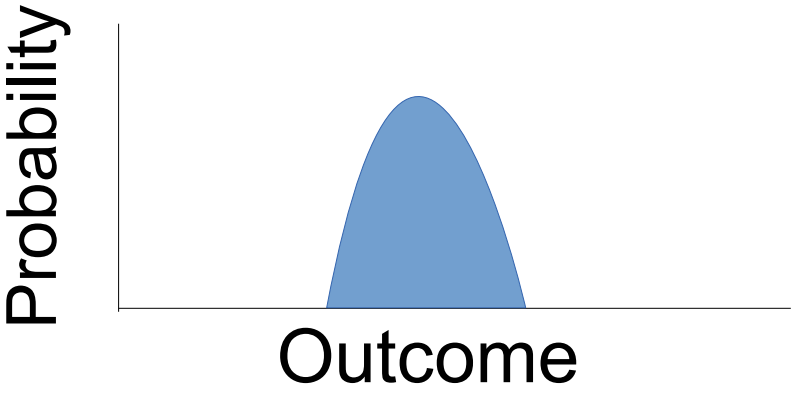
- Nondeterminism
  - But all of these assume that the numbers across benchmarks are valid/deterministic
  - What does nondeterminism do?
  - Every measurement is a sample from a probability distribution



Reporting

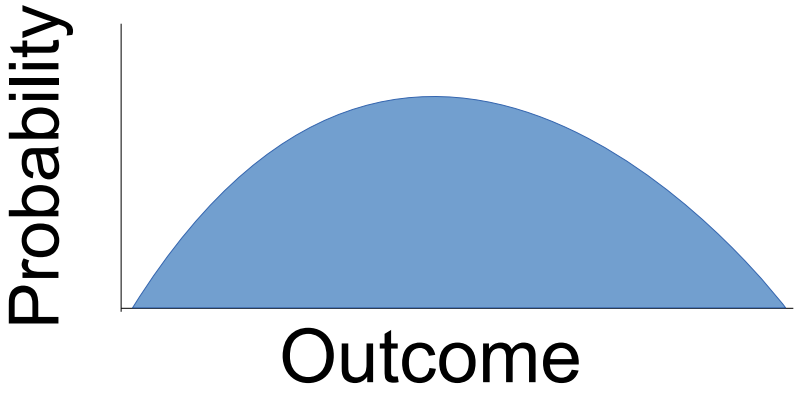
- Do these mean the same thing?

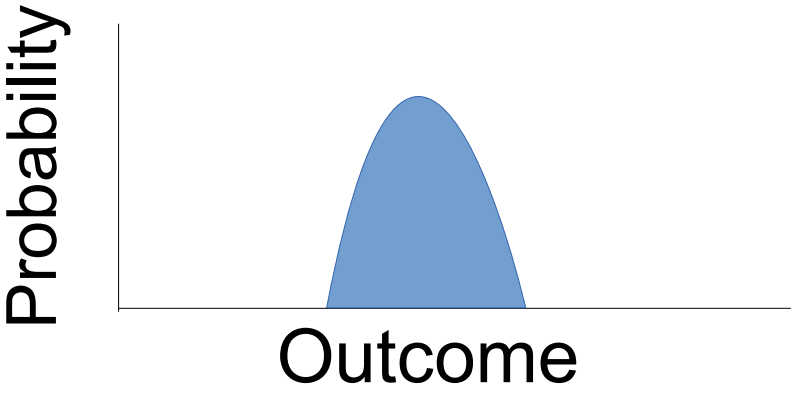




Reporting

- Do these mean the same thing?





You should also worry about the
sample standard deviation

$$\sqrt{\frac{\sum (r_1 - arithmean(r))^2}{N}}$$

Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean

Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If there standard deviation is large:

Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If there standard deviation is large:
    - You may need more samples to build confidence

Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If there standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design

Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If there standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design
- More rigorously, you can even compute
  - Confidence intervals

Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If there standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design
- More rigorously, you can even compute
  - Confidence intervals
    - region in which actual value is likely to be

Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If there standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design
- More rigorously, you can even compute
  - Confidence intervals
  - Confidence tests (e.g. T-tests)

Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If there standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design
- More rigorously, you can even compute
  - Confidence intervals
  - Confidence tests (e.g. T-tests)
    - Test to determine whether the difference between sample means is … meaningful

Reporting

- Standard Deviation
  - Provides a notion of *confidence* in the sample mean
  - If there standard deviation is large:
    - You may need more samples to build confidence
    - There may be methodological error in your test design
- More rigorously, you can even compute
  - Confidence intervals
  - Confidence tests (e.g. T-tests)

    To determine whether results are statistically significant

Reporting

- So what is often done in practice?
  - Firefox(http://arewefastyet.com/)
  - LLVM (http://llvm.org/perf/)