3 Average performance and variability

'The continued fantasy that there is, will be, or should be a single computer architecture for all problem spaces (or a single yardstick to measure such things) continues to fascinate me. Why should computing be different from everything else in Human experience?'

Keith Bierman, in comp.benchmarks

3.1 Why mean values?

The performance of a computer system is truly multidimensional. As a result, it can be very misleading to try to summarize the overall performance of a computer system with a single number. For instance, a computer system may be optimized to execute some types of programs very well. However, this specialization may cause it to perform very poorly when executing a different class of applications. Since the measured execution times of the different classes of applications running on this system will have a very wide range, trying to summarize the performance of this system over all classes of applications using a single mean value can result in very misleading conclusions.

Nevertheless, human nature being what it is, people continue to want a simple way to compare different computer systems. As a result, there continues to be a very strong demand to reduce the performance of a computer system to a single number. The hope is that this single number will somehow capture the essential performance of the system so that comparing performance can be reduced to simply comparing a single mean value for each system. While this is an impossible goal, mean values can be useful for performing coarse comparisons. Furthermore, the performance analyst may be pressured to calculate mean values, and will certainly see others use mean values to justify some result or conclusion. Consequently, it is important to understand how to correctly calculate an appropriate mean value, and how to recognize when a mean has been calculated incorrectly or is being used inappropriately.

As you read this chapter, keep in mind that the computer industry is very competitive, with considerable amounts of money at stake. Each manufacturer

wants their system to have a better performance than their competitors' systems, so they invest a great deal of time and effort in comparing the performances of their system with those of their competitors'. This intense competition pressures them to put the best possible 'spin' on their performance numbers. The seemingly simple question of choosing the correct mean to use, which you would probably assume should be made on purely mathematical grounds, is a good example of the controversy that can develop as a result of these competitive pressures. The discussion of benchmark programs in Chapter 7 will further highlight the pressures performance analysts face to put results in the most favorable light possible.

3.2 Indices of central tendency

The previous chapter pointed out the importance of making several measurements of a program's execution time since the execution time is subject to a variety of nondeterministic effects. The problem then is to summarize all of these measurements into a single number that somehow specifies the center of the distribution of these values. In addition, you may wish to summarize the performance of a system using a single value that is somehow representative of the execution times of several different benchmark programs running on that system. There are three different *indices of central tendency* that are commonly used to summarize multiple measurements: the mean, the median, and the mode.

3.2.1 The sample mean

The *sample arithmetic mean*, or *average*, is the most commonly used measure of central tendency. If the possible values that could be measured are thought of as a random process on the discrete random variable X, the *expected value* of X, denoted $E[X]$, is defined to be

$$E[X] = \sum_{i=1}^{n} x_i p_i \tag{3.1}$$

where p_i is the probability that the value of the random variable X is x_i, and there are n total values. This value is also referred to as the *first moment* of the random variable X.

Using the term 'sample' when discussing the mean value emphasizes the fact that the values used to calculate the mean are but one possible sample of values that could have been measured from the experimental process. This sample mean, denoted $\bar{x}$, is our approximation of the true mean of the underlying

random variable X. This true mean is typically denoted μ. Its true value cannot actually be known since determining this value would require an infinite number of measurements. The best we can do is approximate the true mean with the sample mean. In Chapter 4 we discuss techniques for quantifying how close the sample mean is to the true mean. When there is no chance of confusing whether we mean sample mean or true mean, we simply use the more convenient term 'mean.'

Given n different measurements that we wish to average together, we typically assume that the probabilities of obtaining any of the n values are all equally likely. Thus, our estimate of the sample mean, commonly referred to as the *arithmetic mean*, is

$$\bar{x}_{\mathrm{A}} = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{3.2}$$

As an example of how to calculate a mean, consider the five measurements shown in Table 3.1. The average value is simply the sum of the $n = 5$ measurements divided by n, giving $\bar{x}_A = 15.8$.

3.2.2 The sample median

By design, one of the properties of the sample mean is that it gives equal weight to all measurements. As a result, one value that is significantly different from the other values, called an *outlier*, can have a large influence on the computed value of the resulting mean. For example, if we add a sixth measurement with the value 200 to the five measurements in Table 3.1, the new value for the mean is $\bar{x}_A = 46.5$. This value is substantially higher than most of the measurements and does not seem to capture our 'sense' of the central tendency of the six measurements.

The *median* is an index of central tendency that reduces the skewing effect of outliers on the value of the index. It is found by first ordering all of the n measurements. The middle value is then defined to be the median of the set of measurements. If n is even, the median is defined to be the mean of the middle two values. Using this definition, the median of the five values in Table 3.1 is 16. If the sixth measurement of 200 is also included in this set of measurements, the median becomes the mean of x_4 and x_5 which is 17. So, while adding the sixth value to the set of measurements increases the mean from 15.8 to 46.5, the median increases only from 16 to 17. Thus, given the large outlier in these measurements, the median appears to more intuitively capture a sense of the central tendency of these data than does the mean.

Table 3.1. Sample execution-time measurements used to demonstrate the calculation of the mean and median

Measurement	Execution time
x_1	10
x_2	20
x_3	15
x_4	18
x_5	16

3.2.3 The sample mode

The *mode* is simply the value that occurs most frequently. Note that the mode need not always exist for a given set of sample data. In the example data of Table 3.1, no one value occurs more than once, so there is no mode. Furthermore, the mode need not be unique. If there are several x_i samples that all have the same value, for instance, there would be several modes, specifically each of those x_i sample values.

3.2.4 Selecting among the mean, median, and mode

One nice property of the arithmetic mean is that it gives equal weight to all of the measured values. As a result, it incorporates information from the entire sample of data into the final value. However, this property also makes the mean more sensitive to a few outlier values that do not cluster around the rest of the samples. The median and mode, on the other hand, do not efficiently use all of the available information, but, as a result, they are less sensitive to outliers. So the question becomes that of which index of central tendency is most appropriate for a given situation. The answer to this question lies in the type of data being analyzed, and in its general characteristics.

Categorical data are those that can be grouped into distinct types or categories. For example, the number of different computers in a organization manufactured by different companies would be categorical data. The mode would be the appropriate index to use in this case to summarize the most common type of computer the organization owns. The mean and median really do not make sense in this context.

If the sum of all measurements is a meaningful and interesting value, then the arithmetic mean is an appropriate index. The sum of all of the values shown in Table 3.1 is the total time required to execute all five of the programs tested,

which is an interesting and meaningful value. Thus, the mean of these measurements is also meaningful. However, the sum of the MFLOPS ratings that could be calculated using these execution times is not a meaningful value. Consequently, it is inappropriate to calculate an arithmetic mean for MFLOPS (this issue is discussed further in Section 3.3.2).

Finally, if the sample data contain a few values that are not clustered together with the others, the median may give a more meaningful or intuitive indication of the central tendency of the data than does the mean. As an example, assume that we wish to determine how much memory is installed in the workstations in our laboratory. We investigate and find that 25 machines contain 16 MBytes of memory, 38 machines contain 32 Mbytes, four machines contain 64 Mbytes, and one machine contains 1024 Mbytes. The sum of these values is the total amount of memory in all of the machines, which is calculated to be 2,896 Mbytes. Since this sum is a meaningful value by itself, the mean value of 42.6 Mbytes per machine is also a meaningful value. However, 63 of the 68 machines have 32 Mbytes of memory or less, making the mean value somewhat misleading. Instead, the median value of 32 Mbytes gives a value that is more indicative of the 'typical' machine.

3.3 Other types of means

To complicate matters further, once we have decided that the mean is the appropriate index of central tendency to use for the current situation, we must decide which *type* of mean to use! So far we have discussed the arithmetic mean, but, in fact, there are two other means that are commonly used to summarize computer-systems performance – the harmonic mean and the geometric mean. Unfortunately, these means are sometimes used incorrectly, which can lead to erroneous conclusions.

3.3.1 Characteristics of a good mean

It is possible to apply the formulas described below to calculate a mean value from any set of measured values. However, depending on the physical meaning of these measured values, the resulting mean value calculated need not make any sense. In particular, as discussed in Chapter 2, there are several characteristics that are important for a good performance metric. Since a mean value is calculated directly from the more basic performance metrics described in Chapter 2, any such mean value should also satisfy all of those characteristics.

For instance, if time values are to be averaged together, then the resulting mean value should be *directly proportional* to the total weighted time. Thus, if the

total execution time were to double, so would the value of the corresponding mean, as desired. Conversely, since a rate metric is calculated by dividing the number of operations executed by the total execution time, a mean value calculated with rates should be *inversely proportional* to the total weighted time. That is, if the total execution time were to double, the value of the corresponding mean of the rates should be reduced to one-half of its initial value. Given these basic assumptions, we can now determine whether the arithmetic mean, geometric mean, and harmonic mean produce values that correctly summarize both execution times and rates.

Throughout the following discussion, we assume that we have measured the execution times of n benchmark programs[1] on the same system. Call these times T_i, $1 \leq i \leq n$. Furthermore, we assume that the total work performed by each of the n benchmark programs is constant. Specifically, we assume that each benchmark executes F floating-point operations. This workload then produces an execution rate for benchmark program i of $M_i = F/T_i$ floating-point operations executed per second. We relax this constant-work assumption in Section 3.3.5 when we discuss how to calculate weighted means.

3.3.2 The arithmetic mean

As discussed above, the arithmetic mean is defined to be

$$\bar{x}_{\mathrm{A}} = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{3.3}$$

where the x_i values are the individual measurements being averaged together. In our current situation, $x_i = T_i$ so that the mean execution time is

$$\overline{T}_{\mathrm{A}} = \frac{1}{n}\sum_{i=1}^{n} T_i. \tag{3.4}$$

This equation produces a value for $\overline{T}_{\mathrm{A}}$ that is directly proportional to the total execution time. Thus, the arithmetic mean is the correct mean to summarize execution times.

If we use the arithmetic mean to summarize the execution rates, we find

$$\overline{M}_{\mathrm{A}} = \frac{1}{n}\sum_{i=1}^{n} M_i = \sum_{i=1}^{n} \frac{F/T_i}{n} = \frac{F}{n}\sum_{i=1}^{n} \frac{1}{T_i}. \tag{3.5}$$

[1] A *benchmark program* is any program that is used to measure the performance of a computer system. Certain programs are sometimes defined as a standard reference that can be used for comparing performance results. See Chapter 7 for more details.

This equation produces a result that is directly proportional to the sum of the inverse of the execution times. However, in terms of the characteristics described in Section 3.3.1, we need a value that is inversely proportional to the sum of the times. We conclude, then, that the arithmetic mean is inappropriate for summarizing rates.

3.3.3 The harmonic mean

The second type of mean that is commonly used by performance analysts is the *harmonic mean*. It is defined to be

$$\bar{x}_{\mathrm{H}} = \frac{n}{\sum_{i=1}^{n} 1/x_i} \tag{3.6}$$

where, as before, the x_i values represent the n separate values that are being averaged together.

If we use the harmonic mean to summarize execution-time values, then $x_i = T_i$ and we obtain the following expression:

$$\overline{T}_{\mathrm{H}} = \frac{n}{\sum_{i=1}^{n} 1/T_i}. \tag{3.7}$$

This value is obviously not directly proportional to the total execution time, as required in terms of the properties of a good mean in Section 3.3.1. Thus, we conclude that the harmonic mean is inappropriate for summarizing execution-time measurements.

We find that the harmonic mean is the appropriate mean to use for summarizing rates, however. In this case, $x_i = M_i = F/T_i$, giving

$$\overline{M}_{\mathrm{H}} = \frac{n}{\sum_{i=1}^{n} 1/M_i} = \frac{n}{\sum_{i=1}^{n} T_i/F} = \frac{Fn}{\sum_{i=1}^{n} T_i}. \tag{3.8}$$

This value, which is simply the total number of operations executed by all of the programs measured divided by the sum of all of the execution times, is obviously inversely proportional to the total execution time. Thus, the harmonic mean is appropriate for summarizing rate measurements.

Example. Consider the measurements shown in Table 3.2. The arithmetic mean of the execution times is easily calculated using the sum of the total times. The execution rates are calculated by dividing the total number of floating-point operations executed in each program by its corresponding execution time. The harmonic mean of these rates is then found by calculating the value $\overline{M}_{\mathrm{H}} = 5/(\frac{1}{405} + \frac{1}{367} + \frac{1}{405} + \frac{1}{419} + \frac{1}{388})$. Notice that this value is the same as that obtained by taking the ratio of the total number of floating-point operations executed by all of the programs to the sum of their execution times (within the error due to rounding off). ◇

Table 3.2. An example of calculating the harmonic mean

Measurement (i)	T_i (s)	F (10^9 FLOP)	M_i (MFLOPS)
1	321	130	405
2	436	160	367
3	284	115	405
4	601	252	419
5	482	187	388
$\sum_{i=1}^{5} x_i$	2124	844	
$\overline{T}_{\mathrm{A}}$	425		
$\overline{M}_{\mathrm{H}}$			396

3.3.4 The geometric mean

Some performance analysts have advocated the geometric mean as the appropriate mean to use when summarizing normalized numbers. In fact, it is the mean that is used to summarize the normalized execution times measured in the SPEC benchmark (see Section 2.3.4). It also has been suggested that it is the most appropriate mean to use when summarizing measurements with a wide range of values since a single value has less influence on the geometric mean than it would on the value of the arithmetic mean.

The geometric mean is defined to be the nth root of the product of the n individual x_i values. That is,

$$\bar{x}_{\mathrm{G}} = \sqrt[n]{x_1 x_2 \cdots x_i \cdots x_n} = \left(\prod_{i=1}^{n} x_i \right)^{1/n}. \tag{3.9}$$

Unfortunately, as we will see below, the geometric mean is not an appropriate mean to summarize either times or rates, irrespective of whether they are normalized.

Proponents of the geometric mean say that one of its key advantages is that it maintains consistent relationships when comparing normalized values regardless of the basis system used to normalize the measurements. To test this assertion, we compare the performance of three different computer systems when executing five different benchmark programs. The programs are run on the different systems, producing the execution-time measurements shown in Table 3.3. Using the geometric mean of these measurements to compare these systems shows that S_3 performs the best, followed by S_2 and S_1, in that order. Normalizing the measurements using S_1 as the basis produces the same rank ordering of systems, as

Table 3.3. Execution times of five benchmark programs executed on three different systems

Program	S_1	S_2	S_3
1	417	244	134
2	83	70	70
3	66	153	135
4	39,449	33,527	66,000
5	772	368	369
Geometric mean	587	503	499
Rank	3	2	1

shown in Table 3.4. Similarly, Table 3.5 shows that the same ordering is again preserved when all of the measurements are normalized relative to system S_2.

Unfortunately, although the geometric mean produces a consistent ordering of the systems being compared, it is the wrong ordering. Table 3.6 shows the sums of the execution times of the benchmark programs for each system along with the arithmetic means of these execution times. When these times are used to rank the performances of the three different systems, we see that S_2 performs the best; that is, it produces the shortest execution time, followed by S_1 and then S_3. Since the execution time is the measure of performance in which we are ultimately most interested, it is apparent that the geometric mean produced the wrong ordering. We conclude that, although the geometric mean is consistent regardless of the normalization basis, it is consistently wrong.

It is easy to see why the geometric mean produces the wrong ordering when it is used to average together execution times. In this case, $x_i = T_i$, and

$$\overline{T}_{\mathrm{G}} = \left(\prod_{i=1}^{n} T_i \right)^{1/n}. \tag{3.10}$$

This value is obviously not directly proportional to the total execution time. Similarly, averaging together execution rates with the geometric mean produces

$$\overline{M}_{\mathrm{G}} = \left(\prod_{i=1}^{n} M_i \right)^{1/n} = \left(\prod_{i=1}^{n} \frac{F}{T_i} \right)^{1/n} \tag{3.11}$$

which is not inversely proportional to the total execution time. Both $\overline{T}_{\mathrm{G}}$ and $\overline{M}_{\mathrm{G}}$ violate the characteristics of a good mean value, forcing the conclusion that the geometric mean is inappropriate for summarizing both execution times and rates, irrespective of whether they are normalized.

Table 3.4. The execution times of the benchmark programs in Table 3.3 normalized with respect to that of S_1

Program	S_1	S_2	S_3
1	1.0	0.59	0.32
2	1.0	0.84	0.85
3	1.0	2.32	2.05
4	1.0	0.85	1.67
5	1.0	0.48	0.45
Geometric mean	1.0	0.86	0.84
Rank	3	2	1

Table 3.5. The execution times of the benchmark programs in Table 3.3 normalized with respect to that of S_2

Program	S_1	S_2	S_3
1	1.71	1.00	0.55
2	1.19	1.00	1.00
3	0.43	1.00	0.88
4	1.18	1.00	1.97
5	2.10	1.00	1.00
Geometric mean	1.17	1.00	0.99
Rank	3	2	1

Table 3.6. The total and average execution times of the benchmark programs in Table 3.3.

Program	S_1	S_2	S_3
1	417	244	134
2	83	70	70
3	66	153	135
4	39,449	33,527	66,000
5	772	368	369
Total time	40,787	34,362	66,798
Arithmetic mean	8157	6872	13,342
Rank	2	1	3

3.3.5 Weighted means

The above definitions for the arithmetic and harmonic means implicitly assume that each of the n individual measurements being averaged together is equally important in calculating the mean. In many situations, however, this assumption need not be true. For instance, you may know that half of the time you use your computer system you are running program 1, with the remaining time split evenly between four other application programs. In this case, then, you would like the mean value you calculate to reflect this mix of application-program usage.

This type of *weighted mean* can easily be calculated by assigning an appropriate fraction, or *weight*, to the measurement associated with each program. That is, a value w_i is assigned to program i such that w_i is a fraction representing the relative importance of program i in calculating the mean value, and

$$\sum_{i=1}^{n} w_i = 1. \tag{3.12}$$

In the situation mentioned above, program 1 is used half of the time, so $w_i = 0.5$. The other four programs are used equally in the remaining half of the time, giving $w_2 = w_3 = w_4 = w_5 = 0.125$. Given these weights, the formula for calculating the arithmetic mean becomes

$$\bar{x}_{\mathrm{A,w}} = \sum_{i=1}^{n} w_i x_i \tag{3.13}$$

and the harmonic mean becomes

$$\bar{x}_{\mathrm{H,w}} = \frac{1}{\sum_{i=1}^{n} w_i / x_i}. \tag{3.14}$$

We ignore the geometric mean in this discussion since it is not an appropriate mean for summarizing either execution times or rates.

3.4 Quantifying variability

While mean values are useful for summarizing large amounts of data into a single number, they unfortunately hide the details of how these data are actually distributed. It is often the case, however, that this distribution, or the *variability* in the data, is of more interest than the mean value.

A *histogram* is a useful device for displaying the distribution of a set of measured values. To generate a histogram, first find the minimum and maximum values of the measurements. Then divide this range into b subranges. Each of

these subranges is called a histogram *cell* or *bucket*. Next, count the number of measurements that fall into each cell. A plot of these counts on the vertical axis with the cells on the horizontal axis in a bar-chart format is the histogram. It is also possible to normalize the histogram by dividing the count in each cell by the total number of measurements. The vertical axis then represents the fraction of all measurements that falls into that cell.

One difficulty in constructing a histogram is determining the appropriate size for each cell. There is no hard and fast rule about the range of values that should be grouped into a single cell, but a good rule of thumb is that the width of the cells should be adjusted so that each cell contains a minimum of four or five measurements. (This rule of thumb comes indirectly from our typical assumptions about the distribution of measurement errors, which is discussed in Chapter 4.)

Example. Consider an experiment in which the performance analyst measures the sizes of messages sent on two different computer networks. The average message size for network A was calculated to be 14.9 kbytes, while the average for network B was found to be 14.7 kbytes. On the sole basis of these mean values, the analyst may conclude that the characteristics of the message traffic carried on each network are roughly similar. To verify this conclusion, the message-size measurements are grouped into histogram cells, each with a width of 5 kbytes, as shown in Table 3.7. That is, the first cell is the number of messages within the range 0–5 kbytes, the second cell counts the number of messages within the range 5–10 kbytes, and so forth. As shown in the plots of these two histograms in Figures 3.1 and 3.2, the messages on the two networks have completely different distributions, even though they have almost identical means. ◇

This example demonstrates the problem with relying on a single value to characterize a group of measurements. It also shows how the additional detail in a histogram can provide further insights into the underlying system behavior. However, while the two histograms in this example are obviously substantially different, visually comparing two histograms can be imprecise. Furthermore, histograms can often provide too much detail, making it difficult to quantitatively compare the spread of the measurements around the mean value. What is needed, then, is a single number that somehow captures how 'spread out' the measurements are. In conjunction with the mean value, this *index of dispersion* provides a more precise metric with which to summarize the characteristics of a group of measurements. The question then becomes one of choosing an appropriate metric to quantify this dispersion.

Perhaps the simplest metric for an index of dispersion is the *range*. The range is found by taking the difference of the maximum and minimum of the measured values:

Table 3.7. The number of messages of the indicated sizes sent on two different networks

Message size (kbytes)	Network A	Network B
$0 < x_i \leq 5$	11	39
$5 < x_i \leq 10$	27	25
$10 < x_i \leq 15$	41	18
$15 < x_i \leq 20$	32	5
$20 < x_i \leq 25$	21	19
$25 < x_i \leq 30$	12	42
$30 < x_i \leq 35$	4	0

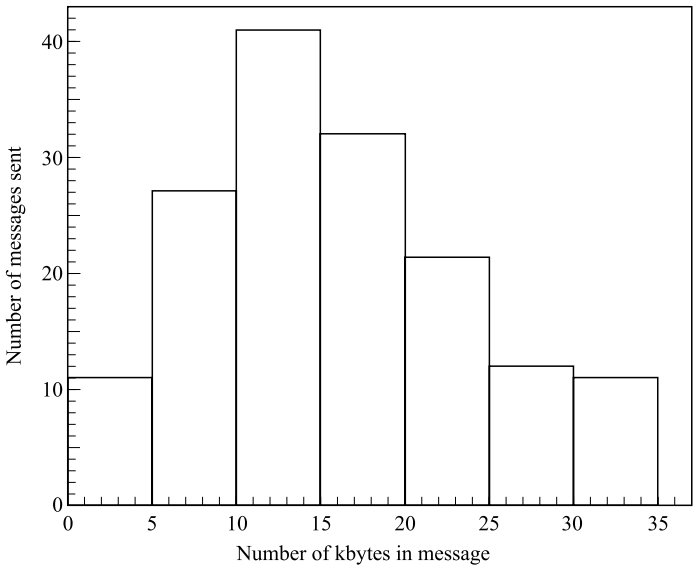

Figure 3.1 A histogram plot of the data for network A from Table 3.7.

$$R_{\max} = \max_{\forall i} x_i - \min_{\forall i} x_i. \tag{3.15}$$

Although it is simple to calculate, the range does not use all of the available information in summarizing the dispersion. Thus, it is very sensitive to a few extreme values that need not be representative of the overall set of measurements. A slight improvement is to find the maximum of the absolute values of the difference of each measurement from the mean value:

$$\Delta_{\max} = \max_{\forall i} |x_i - \bar{x}|. \tag{3.16}$$

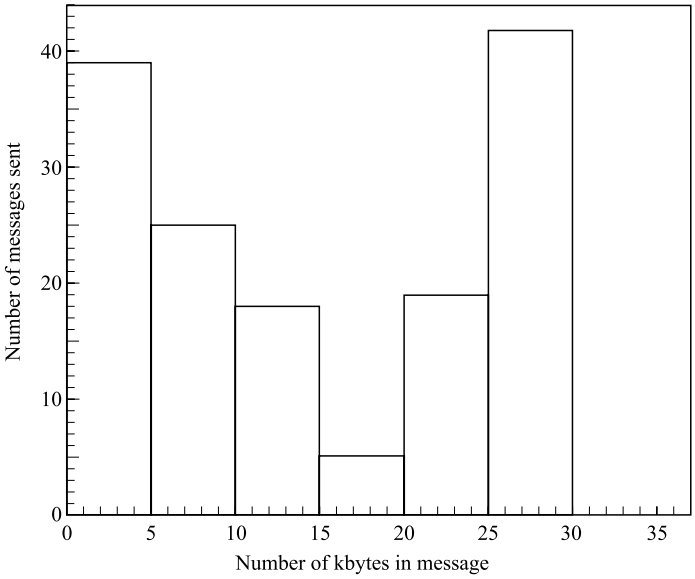

Figure 3.2 A histogram plot of the data for network B from Table 3.7.

Again, however, this value does not efficiently take advantage of all of the available information, and is overly sensitive to extreme values.

A better, and perhaps the most commonly accepted, index of dispersion is the variance. The *sample variance* is our calculated estimate of the actual variance of the underlying distribution from which our measurements are taken. It incorporates all of the information available about the difference of each measurement from the mean value. It is defined to be

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} \tag{3.17}$$

where the x_i are the n independent measurements, and $\bar{x}$ is the corresponding arithmetic mean. Notice in this equation that only $n-1$ of the differences $x_i - \bar{x}$ are independent. That is, the nth difference, $x_n - \bar{x}$, could be computed given the other $n-1$ differences. Thus, the number of *degrees of freedom* in this equation, which is the number of independent terms in the sum, is $n-1$. As a result, the sum of the squared differences in this equation is divided by $n-1$ instead of n.

This equation defines the sample variance, but it is not particularly useful for calculating the variance given a set of measurements. Furthermore, this definition requires our knowing the mean value, $\bar{x}$, before calculating the variance. This implies that two passes must be made through the data, once to calculate the mean and a second pass to find the variance. This requirement makes it difficult to calculate the variance 'on the fly' as the data are being generated,

for instance. To facilitate calculating the variance, we can expand Equation (3.17) to give

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i^2 - 2\bar{x}x_i + \bar{x}^2) = \frac{n\sum_{i=1}^{n}x_i^2 - (\sum_{i=1}^{n}x_i)^2}{n(n-1)}. \tag{3.18}$$

This equation shows that, to calculate the variance, we need to make only a single pass through the data to find the sum of the x_i values and the sum of the x_i^2 values. We can then use these sums to calculate both the mean and the variance.

One of the problems in using the variance to obtain an indication of how large the dispersion of data is relative to the mean is that the units of the variance are the square of the units of the values actually measured. In the above example, for instance, the units of the individual measurements, and so, therefore, of the mean, are bytes. The units of the variance, however, are bytes squared. This squared relationship of the units of the variance to those of the mean makes it difficult to compare the magnitude of the variance directly with the magnitude of the mean.

A more useful metric for this type of comparison is the *standard deviation*, which is defined as the positive square root of the variance. That is, the sample standard deviation is

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}. \tag{3.19}$$

With this definition, the mean and standard deviation have the same units, making comparisons easier. Finally, use of the *coefficient of variation* (COV) eliminates the problem of specific units by normalizing the standard deviation with respect to the mean. The coefficient of variation is defined to be

$$COV = s/\bar{x} \tag{3.20}$$

and so provides a dimensionless value that compares the relative size of the variation in the measurements with the mean value of those measurements.

3.5 Summary

Several different types of means can be used to summarize a collection of measurements with a single number. Although this summarization hides much of the information provided by the n different measurements, human nature persists in wanting to reduce performance to a single number to simplify the task of making

comparisons. Consequently, it is important for the performance analyst to understand the definitions of the different means, and how to use each appropriately. The following points summarize how to select an appropriate mean for a given situation.

- **The arithmetic mean.** The arithmetic mean is the appropriate choice whenever the sum of the raw results has some physical meaning and is an 'interesting' value. For example, the sum of execution times is a total execution time, which is both meaningful and interesting. Similarly, the total number of bytes sent by messages on a communications network has physical meaning and by itself is an interesting value. The arithmetic mean should *not* be used to summarize rates.
- **The harmonic mean**. The harmonic mean is the appropriate mean for summarizing rates since it reduces to the total number of operations executed by all of the test programs divided by the total time required to execute those operations, which is simply the definition of the total execution rate. It is not appropriate to use the harmonic mean to summarize measurements that should be summarized using the arithmetic mean, such as execution times.
- **The geometric mean**. Although it has been advocated as the best mean to use for summarizing normalized values, the geometric mean is not appropriate for summarizing either rates or times, irrespective of whether they are normalized.
- **Normalization**. Owing to the mathematical difficulties of averaging together normalized values, it is best to first calculate the appropriate mean and then perform the desired normalization.

In addition to these mean values, we introduced the median and the mode as other measures of central tendency. As the middle value in a collection of measurements, the median is useful when the measurements have a few outlying values that tend to distort the intuitive sense of the measurement's central tendency. The mode is useful for quantifying the most common value among a set of categorical measurements.

One of the problems with these single-value summaries of a collection of measurements is that they hide their variability. A histogram is a useful graphical representation for displaying this variability. The variance (or the standard deviation) is a statistic that can be used to summarize in a single number the variability shown in a histogram.

3.6 For further reading

- This paper describes the three types of means and argues for the use of the geometric mean for averaging normalized numbers:

P. J. Fleming and J. J. Wallace, 'How Not To Lie With Statistics: The Correct Way To Summarize Benchmark Results,' *Communications of the ACM*, Vol. 29, No. 3, March 1986, pp. 218–221.

- The following paper, however, argues against the use of the geometric mean. It also introduces several of the ideas of what constitutes a good mean that were presented in this chapter:

  James E. Smith, 'Characterizing Computer Performance with a Single Number,' *Communications of the ACM*, October 1988, pp. 1202–1206.

  Taken together, these two papers provide an interesting glimpse into the controversy that can arise among performance analysts over such fundamental concepts as selecting an appropriate mean with which to summarize a set of measured values.

- Almost any introductory statistics text will provide a development of the basic types of means and the variance.

3.7 Exercises

1. What aspects of a computer system's performance is it reasonable to summarize with a single number?
2. It has been said (Smith, 1988) that the geometric mean is consistent, but it is consistently wrong. A mean is calculated according to a well-defined formula, so in what sense can it be wrong?
3. Which measure of central tendency, the mean, median, or mode, should be used to summarize the following types of data: size of messages in a communication network, number of cache hits and misses, execution time, MFLOPS, MIPS, bandwidth, latency, speedup, price, image resolution, and communication throughput? For those for which the mean is the best choice, which mean should be used (arithmetic, geometric, or harmonic)?
4. Table 3.8 shows the execution times measured for several different benchmark programs when they are executed on three different systems. The last column shows the number of instructions executed by each of the benchmark programs. Assuming that each benchmark should be equally weighted, calculate the following values:
   (a) the average execution time,
   (b) the average MIPS rate, and
   (c) the average speedup and relative change when using S_3 as the basis system.

Table 3.8. The times measured on several different systems for a few benchmark programs

Program	S_1	S_2	S_3	Number of instructions
1	33.4	28.8	28.3	1.45×10^{10}
2	19.9	22.1	25.3	7.97×10^{9}
3	6.5	5.3	4.7	3.11×10^{9}
4	84.3	75.8	80.1	3.77×10^{10}
5	101.1	99.4	70.2	4.56×10^{10}

(d) Are these average values reasonable summaries of the data presented? Why or why not?

5. Repeat the above problem when benchmark program 1 represents 40% of the expected workload, benchmark program 2 35%, benchmark program 3 15%, and benchmark programs 4 and 5 each 5%.
6. Determine the coefficient of variation of the execution times for each system shown in Table 3.8.