The Infamous Ratio Measure

Hakan Erdogmus

Software productivity and quality are commonly expressed as derived metrics—in particular, the ratio of two base measures. For productivity, which I addressed in my previous column (“Measurement Acquiescence,” March/April 2008), the base measures are output (or a proxy thereof) and effort. For quality, common base measures are defect count and software size. The March/April 2007 IEEE Software article “Misleading Metrics and Unsound Analyses” (by Barbara Kitchenham, David Ross Jeffery, and Colin Connaughton) drew attention to the pitfalls of using ratios; however, ratios and other derived metrics are inevitable in software measurement. An appreciation of what they’re good for, and not good for, can help to avoid these pitfalls. I focus on ratios here.

Why we need ratios
We need ratios for whatever limited portability we can achieve in software measurement.

Ratios provide the relativity required for comparison, something we can’t do with only absolute measures. They increase portability by hiding scale. Ironically, the loss of scale information also counteracts portability in various decision-making situations in which economies and diseconomies of scale play an important role. But without abstracting away from scale at least temporarily, it’s difficult to talk about how projects, artifacts, and teams behave over time, both relative to each other and in response to changes in context.

Economic concepts such as productivity are inherently defined as ratios. Dependencies among base variables should not modify such definitions. For example, the effort prediction model COCOMO (Constructive Cost Model) suggests a polynomial relationship between effort and software size, resulting in diseconomies of scale. However, would this dependency change the COCOMO definition of software productivity from something other than the simple ratio of size to effort? The information, according to COCOMO, that larger projects tend to be less productive is external to this ratio metric.

The tendency of projects to move at different rates as they progress is observable if we track their absolute output over a moving time window of fixed duration. However, if the observations’ durations vary significantly, we must resort to using ratios by normalizing absolute measurements with respect to individual observations’ scopes. Similarly, if resources vary over time, meaningful comparison is possible if the measurements are adjusted for differences in scale. Such comparisons need not, and often should not, be for the express purpose of deciding which projects, artifacts, or teams are “better” overall performers. They may simply reveal notable, and justified, trends in project and product outcomes.

Interpreting simple ratios
A ratio metric’s proper interpretation begins with acknowledging the loss of information originally captured in the base measures. Consider a sprinter’s and a marathoner’s performance over several races

Mission Statement: To be the best source of reliable, useful, peer-reviewed information for leading software practitioners—the developers and managers who want to keep up with rapid technology change.
of various distances. The marathoner’s average speed, a ratio metric, would be lower than the sprinter’s. This information isn’t enough to decide which runner is a more accomplished athlete or estimate who’s likely to cover an arbitrarily specified distance in the shortest possible time. However, it lets us set behavioral expectations for specific decisions. If we’re considering hiring one of the runners as a messenger for short-distance delivery services subject to strict deadlines, knowing that the sprinter is faster (based on average speed) over repeated dashes (based on the context in which that level of performance is valid), we might choose the sprinter.

Think of a ratio metric as representing an instantaneous quantity, a rate of change, or an average as opposed to an invariant equation relating the metric’s value to its components. Remember that the metric’s value is not independently and directly observable: it can be computed only from direct observations of the base measures. We must therefore avoid assuming a linear model governing the relationship between the metric and the base measures. A derived metric’s value usually cannot be controlled by arbitrarily manipulating the base measures. A case in point is defect density, a familiar ratio metric used to gauge software quality. We can’t achieve a smaller defect density simply by increasing a system’s size. Suppose an ongoing project has a defect density of 1.5 defects per KLOC at its current size of 10 Kbytes. On the basis of this measurement, should we expect about 15 defects when the software eventually grows to 100 KLOC? A “yes” answer would mistakenly imply a linear predictive model. Here defect density represents a short-term measurement without long-term predictive power.

**Spurious correlations**

When I started looking more closely at the implications of using ratios, a colleague referred me to the book *Ratio Correlation: A Manual for Students of Petrology and Geochemistry* (University of Chicago Press, 1971). In this book, Felix Chayes presents statistical techniques for detecting relationships between molecules found in earth samples. In petrology and geochemistry, the occurrence frequency of molecules of interest is expressed as proportions because absolute counts aren’t directly comparable across samples. Correlations between such proportions are the main focus of Chayes’ book.

A proportion is the ratio of two quantities where each quantity is the sum of a set of base variables. When two proportions (say X/(X + Y) and Y/(Y + Z)) share base variables (in this case, Y), they exhibit a spurious correlation, called a null correlation, due to the shared component. Therefore, any statistical test designed to reveal a true relationship between two proportions must take into account this null correlation, for otherwise the test might indicate a false association. This insight also applies to a software metric expressed as a ratio.

**Predictive models**

Suppose a large legacy system’s modules are randomly sampled for testing and the sampled modules’ defect counts are tracked to optimally allocate testing resources. Based on multiple observations, we wish to develop a predictive model to estimate the defect density of an arbitrary module from the module’s size. Should we regress module size against defect density? Or is it better to regress module size directly against the number of defects found and then estimate defect density from the predicted defect count?

Figure 1 shows artificially generated scatter plots of defect density versus module size for uncorrelated and correlated base measure samples. Distinguishing between the two plots is difficult because of the negative null correlation’s domination. Hence, any predictive model derived from these measurements must take care to eliminate the spurious effect. Figure 2, on the other hand, shows plots of the absolute number of defects found against module size for the same data, where the relationship between size and quality is explicit. A predictive model is thus better off using the base variables directly. In this case, the use of the ratio metric, defect density, as a dependent variable would be unwise because it artificially increases the strength of the underlying relationship.

**Functional forms**

The scatter plots in figure 1 don’t give us much information about the functional relationship between size and quality. It’s apparent that size and quality are inversely proportional; however, this insight isn’t particularly surprising or useful. But the
Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.

Some experts justifiably stress that functional relationships are more important than correlation measures. Weighing defects against size may be too helpful because they obscure the actual functional forms. However, if the goal is to understand behavior better, simple ratios might not be proportionately less defect prone, with implications for distributing limited testing resources among differently sized modules.
One temporary solution is to adjust a simple ratio metric for any observed scale economies or diseconomies so that values represent something closer to a true ratio scale. For figure 2b, such a metric would be defects squared per unit size (the observations were generated using a quadratic function with a random error). Unfortunately, this solution risks being transient, and therefore impractical. Although functional forms matter, we don’t know yet how fluid or stable they are. Are they universal enough to be treated as laws of nature? Until empirical evidence demonstrates that new technologies and processes do not modify these forms, it would be safer to acknowledge the possibility that functional associations between size and defects as well as effort and output vary both over time and across contexts.

Until then, simple ratios, with their interpretation as first derivatives or averages, will exhibit more universality. Some attempts to discover general, persistent laws may prove fruitful. Notably, A. Güneş Koru and his colleagues Dongsong Zhang and Hongfang Liu have been gathering support for a logarithmic relationship between module size and defect proneness (see their PROMISE 2007 paper “Modeling the Effect of Size on Defect Proneness for Open Source Software” at http://promisedata.org/?cat=130). If such findings are ultimately shown to be insensitive to contextual factors and future innovations, a revision of traditional definitions of software productivity and quality will be warranted.

So, should software measurement avoid simple ratios and other derived metrics? Not as long as we interpret them at face value, use them intelligently, consider scale effects in their interpretation and usage, and don’t discount the value of raw data and contextual information. Derived metrics help us quickly see the big picture, but they are often inadequate, and they can at times be misleading.

If your perspective differs, write to me at hakan.erdogmus@computer.org.

Acknowledgments

Many thanks to A. Güneş Koru and Khaled El-Emam who over the years have illuminated me through engaging discussions.

CALL FOR ARTICLES

**Minning Software Archives**

**Publication:** Jan./Feb. 2009

**Submission Deadline:** 1 July 2008

As we organize and integrate development activities using software tools, we leave behind automatically accessible data. We can glean a tremendous amount of information from these archives by applying data-mining and machine-learning techniques. The resulting patterns, rules, and insights can increase program quality and programmer productivity.

**This special issue seeks articles on**

- Approaches, applications, tools, search techniques, visualization techniques
- Case studies on extracting data from archives of large, long-lived projects
- Analyzing change patterns, collecting and predicting effort data
- Mining social networks from large software projects
- Techniques for predicting reliability and defects
- Mining requirements documents and predicting requirements volatility
- Prioritizing test suites by leveraging historical test data
- How developers, teams, and managers can leverage mined data in day-to-day work

Articles should have a practical orientation and be written in a style accessible to practitioners. Overly complex, purely research-oriented, or theoretical treatments are not appropriate.

**Guest Editors:**

- Nachiappan Nagappan, Microsoft Research, nachin@microsoft.com
- Andreas Zeller, Saarland University, zeller@acm.org
- Tom Zimmermann, University of Calgary, tz@acm.org

For general author guidelines: www.computer.org/software/author.htm
For submission details: software@computer.org