In Search of an Accuracy Metric

(Data and Information Quality Metrics)

Fisher, Craig
Marist College
Craig.Fisher@Marist.edu

Lauria, Eitel
Marist College
Eitel.Lauria@Marist.edu

Matheus, Carolyn
SUNY Albany
ccmatheus@hotmail.com

ABSTRACT

Practitioners and researchers often refer to error rates or accuracy percentages of databases. The former is the number of cells in error divided by the total number of cells; the latter is the number of correct cells divided by the total number of cells. However, databases may have similar error rates (or accuracy percentages) but differ drastically in the severity of their accuracy problems. A simple percent does not provide information as to whether the errors are systematic such as one record with 20 fields in error or 20 errors randomly distributed throughout the database. The difference is rooted in the degree of randomness or complexity. We expand the accuracy metric to include a complexity (randomness) measure and include a probability distribution value. The proposed randomness check is based on the Lempel-Ziv (LZ) complexity measure. The main candidate for the probability distribution parameter is Poisson's lambda. The newly described metric allows management to distinguish between databases that have similar accuracy measures and error rates but differ drastically in the level of complexity of the quality problems.

WHAT IS ACCURACY?

The concept of accuracy is intuitive and it means to be correct or right in a judgment. Data accuracy is thought to be an intrinsic dimension [13] but some say that "… there is no exact definition for accuracy" ([37], p. 93). Others stated that accuracy is not a simple, single quantitative metric [10]. Redman said that "the science of data quality has not yet advanced to the point where there are standard measures for any of the [data quality] issues" [30]. The variety of definitions at the field level include, but are not limited to, the following: the correct mapping of the real world into an information system [13, 22, 37]; correctness of data, implying that there is some way to determine the actual correct number [37]; erroneous values assigned to attributes of some entity [25]; and agreement of an attribute's true value with its recorded value [25].

One could follow the Webster's dictionary that defines accuracy as being free from error [1]. This definition applies to individual cells which are data attributes for specific records. An example of a single cell is the quantity-on-hand field in an inventory record. For example, if the number 1000 was recorded in the quantity field but a physical count found only 990 parts, then it is understood that the recorded value was inaccurate. Accuracy can also refer to being right in judgment by giving correct facts, arguments and reaching proper conclusions. Pierce stated that to determine accuracy at the cell level requires a way to

determine the exact value in the true world [25]. Accuracy for an individual cell is quite different from defining a useful metric for a database--it is not always easy to find the most useful definition.

Going beyond the attribute or field level accuracy for a database has been referred to as "The *extent* to which data are correct, reliable and certified free of error" ([38], p. 31); "judgment of whether the system contains the correct values" [16]; "…the ratio between the number of correct values and the total number of values available from a given source" [6]; proportion of recorded diagnoses that are true [27] and Redman's error rate--the number of erred fields divided by the total number of fields [29, 30]. Lee et al. summarized the general form of rating databases as Rating = 1 – (Number of undesirable outcomes / total outcomes) ([18], p. 54). These definitions have been largely unchallenged but our review indicates a need for a fuller accuracy definition.

REASONS FOR MEASUREMENTS

Management's goal is to improve data and information quality. The Total Quality Management (TQM) literature emphasizes that an important aspect for improving something is being able to measure it. Redman states that which does not get measured does not get managed [30]. The implication is that the measurements can be repeated and monitored over time to determine whether the database is improving or deteriorating. Reimann says that "the most common factor among companies scoring high in the [Baldrige TQM award] evaluation process is that they had instituted systematic measurement processes [28]." A closely related question is when should something be measured?

It is important to be able to benchmark quality. As such, before and after measurements can provide information as to the efficacy of some action in an effort to improve quality. Often, management consider several possible data quality improvement projects and must have a good way to prioritize these projects [5]. In order to evaluate projects management needs to be able to measure current quality and anticipated quality [5]. In addition, it is important to be able to measure quality to determine best-of-breed across a variety of laboratories, plants, branch offices, retail stores and so forth. In the case of multiple actions, efforts will be made to determine which action had the most positive effect, the least amount of cost or some combination thereof. In this case it is desirable to know both and then use some algorithms (such as optimization modeling) to determine the best course of action [5].

Management needs to measure the target databases and systems to determine the relative severity of the data quality problems in those databases. Combined with other factors, such as importance of the application function that uses the database, the number of errors per database or *current quality*, might influence the priority of establishing a project to fix the errors [5]. It is also important to be able to monitor a database on an ongoing basis to be able to detect errors and trends before they become more serious. The most common measurement of accuracy is the simple ratio (actual / estimate) [26] or the number of cells in error / number of cells [18, 29].

In summary, the main purposes of measurement are to monitor trends to determine if a database's quality is deteriorating or improving, to compare the quality of one database to another (e.g., to influence project priorities), to determine best-of-breed and benchmarking to determine if a project to improve quality did in fact improve the quality. Olson adds additional reasons such as negotiating the price, value and usefulness of data with an information vendor and qualifying data before it is applied from one database to another such as a data warehouse [23]. For example, when the vendor says less than 10% errors does the vendor mean random or systematic errors?

DEFICIENCIES IN CURRENT ACCURACY MEASUREMENTS

The widely accepted accuracy (error) rate approaches are deficient. Many articles simply report that a table/file is 90% accurate or contains 10% errors. The deficiency here is based upon whether the errors have systematic root causes or are completely random. This lack of precision contributes to an inability to compare tables/files, to do benchmarking, inhibits monitoring, and makes it difficult to determine best-of-

breed. There are situations where multiple files could all contain 10% errors but have drastically different degrees of accuracy brought about by the distribution of those errors. One can readily see the difference in complexity that is possible. One file might have dozens of records with only a specific column/field in error in each row. This might lead to a very simple fix. Perhaps the program that created the database had one character too little or too much causing a misalignment. A programmer can quickly fix that problem. Another file might report the same percentage of errors but have the errors randomly distributed among many columns, causing both diagnoses and repair to be significantly more involved.

A corollary would be if all errors were congested in just a few rows. This is a drastically different situation than if the errors were randomly distributed across multiple rows and in various columns. Redman's measurement that counts the cells with errors does not capture these differences. All three cases (errors in the same columns, errors in the same rows, or randomly distributed throughout the file) would report the same accuracy percentage but represent significantly different degrees of quality. Consider the 3 tables below.

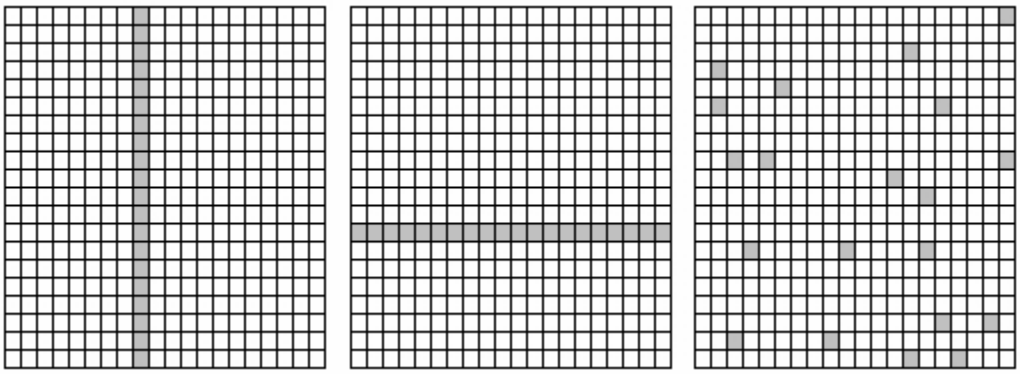

Figure 1. Distribution of Errors
(a) Errors concentrated in one column; (b) Errors concentrated in one row;
(c) Errors distributed throughout the table

The tables exhibited in Figure 1 all have a 5% error rate (20 / 400), using Redman's Cells with errors divided by total number of cells [29]. However, a Quality Analyst assigned to diagnose and repair the tables would most likely deny the equivalence of these tables. The tables in figure 1(a) and 1(b) do not contain a random distribution of errors. The table in Figure 1(a) might be corrected by a single program fix, perhaps by realigning decimal points or by shifting characters to the right or left one space. The table in Figure 1(b) might be fixed by correcting a single input record that was, for example, faxed in late and was so illegible that the data entry person mis-keyed many characters across the whole form. Another possible explanation is a common transmission problem jumbled many bytes in all fields for the single record in error. The table in Figure 1(c) is significantly more complex from a problem solving point of view. There are random errors distributed across many rows and many columns. Most would agree that significantly more effort would be required to fix the table represented in Figure 1(c).

If effort to repair is a good indication then it is clear that the measure of percent of cells with an error divided by the total number of cells is not a clear measure. Due to the aforementioned deficiency, one cannot compare the tables using that measure. The fact that all 3 tables are 95% accurate does not reflect the true state of affairs. Management might rightly judge that the input and file maintenance processes are out of control for the Figure 1(c) case.

Weighting of Field Importance and Severity of Specific Errors

All cells are clearly not created equal. There are several factors that organizations could consider in evaluating accuracy. Severity of errors may vary by type of field in error, type of error and organization. "A key management objective when dealing with information products is to understand the value placed by users on these products" [28]. A few are now mentioned to alert the reader to areas for further study. All metrics that are discussed may make profitable use of applying weights or severity codes to various rows, columns and cells. Organizations might like to associate a high importance weighting factor with a column such as the salary field but a low importance weighting factor with a street name. Some columns might be numeric and more sensitive to errors than other columns.

As several researchers and practitioners alike have said, quality is most effectively defined as fitness for use [4, 13]. Since the usage must be known, accuracy is not always determined by an intrinsic measurement. An example of a specialized usage can be found in Audini, et al. who reviewed the accuracy of Department of Health returns on provision of mental health residential accommodations. In this case, *accuracy* is gauged by the extent to which provisions that should be reported are in fact reported [3]. Accuracy is often confounded with a variety of other dimensions such as timeliness, consistency, and completeness. For example, Audini's accuracy definition closely resembles the definition of completeness in the data quality literature [18].

It would be clearly inaccurate to read the name Smith but write the name Smyth for which there could be some concerns. However, usage would determine the severity. One use for a name might be in everyday mailing of letters across the United States while another use for names might be part of a terrorist tracking database system. If the use was for a mailman to deliver a letter to Mr. Smith at 123 Talburt Street, then that letter would get there just as effectively as if it said Mr. Smyth at 123 Talburt Street. On the other hand, if the name is used in searching a database for Mr. Smith (or Smyth) the system would need some intelligence to provide similar names, much like providing options in a spell checker. The name error mentioned is similar to the FBI's terrorist tracking database wherein many names were misspelled and no one knew how to correct or catch the spelling errors [9]. Other fields that can readily be seen to have different priorities depending upon the usage and organization are salaries, account numbers, employee identification numbers and so forth. Severity and usage weighting will be left for further research.

ACCURACY METRIC

Simply put, the current expansion of the Accuracy-Rate metric is twofold. Our accuracy-metric can be defined as a three parameter vector starting with the well established error rate (accuracy percentage) followed by a complexity measurement and concluding with Poisson's λ or Accuracy = {accuracy-rate, complexity, λ}. The complexity parameter clarifies the degree of difference between systematic patterns versus random distributions of errors. The Lempel-Ziv (LZ) measure of complexity provides information to distinguish between degrees of randomness and therefore complexity. The logic to support LZ as the second parameter will be developed in the next section.

Once it is determined that the errors are distributed randomly, the investigation will turn to the application of an appropriate probability distribution to help management answer important questions such as, "What is the probability that any given record contains zero errors?" or "What is the probability of having between two and four errors per record?" This will be covered in the section entitled Poisson distribution.

RANDOMNESS CHECK

The term 'randomness' is used in an intuitive manner in every day life activities and in professional settings to describe lack of regularity; that is, the lack of an obvious or implicit rule to govern the construction of a pattern. Ordinarily, outcomes generated by a random process are patternless [11]. "Sequences whose order is not random cast a doubt on the random nature of the generating process" [11].

However, randomness is an elusive concept in mathematics [14]. It is complex to define randomness rigorously, and there is no definitive test to establish its presence[1].

The modern definition of randomness has its roots in information theory, a branch of mathematics that studies the storage and transmission of messages. It was originally developed in the late 1940s by Claude Shannon [32]. From an information theory perspective, randomness, complexity and compressibility are closely related concepts. Shannon's noiseless coding theorem showed that for a message given by sequence of symbols S with alphabet $A \equiv \{a_1, a_2, .., a_m\}$ generated independently with probabilities $p(a_i)$, the optimal code has a length in bits of $H(S) = -\sum_i p(a_i) \log_2 p(a_i)$, called the entropy of S. This clearly establishes a one-to-one relationship between the probability of a message and the description length of its optimal code, where short description lengths correspond to large probabilities and vice versa. Borrowed from thermodynamics, Shannon used the term entropy to describe how much information (measured in bits) is encoded in a message composed of a sequence of symbols. The higher the entropy of the message the more information it contains. Messages comprising a simple pattern can be compressed into short descriptions. This is not the case of random messages that require longer code descriptions. Following an example by Chaitin [7], consider these two sequences of binary digits:

0101010101010101010101
01101100110111100010

The first is based on a simple rule: repeat the sequence 01 ten times. The second sequence of symbols has no underlying pattern; the sequence appears to be a random assortment of 0's and 1's. Intuitively, we attribute such a lack of pattern as being random, and we use the term "random sequence" synonymously with "incompressible sequence" [21].

The realization that the length of the minimal description that encodes a message conveys the complexity of the message has led to a new characterization of randomness[2]. Different attempts have been applied to conceptualize the complexity of strings of symbols based on its minimal description. The Kolmogorov (aka algorithmic) complexity of a sequence of binary digits is defined as the shortest computer program for a universal Turing machine that can output the sequence of bits [7, 17, 33]. As stated by Chaitin, "a series of numbers is random if the smallest algorithm capable of specifying it to a computer has about the same number of bits of information as the series itself"[7]. Rooted in the research by Kolmogorov and others, Rissanen (1978) worked on the formulation of stochastic complexity and the minimum description length (MDL) principle, equating the description of the data by a statistical model to the encoding of a message[3]. Thus, the purpose of modeling is to find models that allow short encodings

---

[1] "A given number cannot be proved to be random. This enigma establishes a limit to what is possible in mathematics 7. Chaitin, G.J., *Randomness and mathematical proof.* Scientific American, 1975. **232**(5): p. 47-52."

[2] This is another way of describing Occam's razor, the inductive bias of science to model selection that can be summarized as "choose the shortest explanation for the observed data"

[3] The MDL principle can be analyzed from a Bayesian point of view. If we take the posterior probability $p(M \mid D)$ of a model M based on dataset D, according to Bayes theorem, $p(M \mid D) \propto p(D \mid M) \cdot p(M)$ .erally accepted criterion is that the best model M^* is the one that maximizes $p(M \mid D)$; in other words, $M^* = \arg\max \left[p(D \mid M) \cdot p(M) \right]$. By applying logarithms and inverting the sign of the aforementioned expression, we get $M^* = \arg\min \left[-\log p(D \mid M) - \log p(M) \right]$, which can be interpreted in the following way: select the model M^* which minimizes the sum of the bit-length of the description of the model (L_M) and the bit-length of the description of the data when encoded by the model ($L_{D|M|}$). In other words, $M^* = \arg\min \left[L_{D|M|} + L_{M|} \right]$.

of the data and therefore compress the data. Solomonoff, Kolmogorov and Chaitin's work showed that the complexity (i.e. randomness) of a given binary sequence is the shortest computer program that can reproduce this sequence, but the problem is that a general algorithm for such a program cannot be given.

In the same spirit, but following a slightly different approach, Lempel and Ziv restricted the scope of such programs to one class, allowing two operations on symbols: copy and insert [15, 19]. Instead of computing the length of such program, they developed a complexity measure "related to the number of steps in a self-delimiting production process by which a given sequence is presumed to be generated" and "to the number of distinct substrings and the rate of their occurrence along the sequence" [19]. Lempel and Ziv's algorithm parses the sequence on *n* binary digits into consecutive disjointed substrings such that the next substring is the shortest pattern not found before. This number *c(n)* of disjoint strings has been shown to be an appropriate measure of the Kolmogorov complexity [19].

To illustrate this procedure, consider the sequence of nine (n=9) symbols S= 010110011. Lempel and Ziv's algorithm parses the sequence into five substrings {0,1,01,10,011} rendering $c(n=9)= 5$. The algorithm is simple and is best explained by applying it step by step to sequence S. We will use the notation S(i) to identify the *i*th bit in S. The algorithm parses S from left to right looking for substrings that are not present in the vocabulary **V**. As the algorithm proceeds and the vocabulary is grown dynamically, the beginning of the vocabulary **V** is empty (**V** = ∅). The first substring seen from left to right is S(1)=0, and **V**={0}. We parse S(2)=1 and add it to the **V**. Thus far **V**={0,1}. The next bit is S(3)=0, already present in **V**, so we append S(4)=1 to it, rendering substring 01. As 01 is not present in the vocabulary, we include it: **V**={1,0,01}. The next bit S(5)=1 is included in **V**, so we append S(6)=0 to it. The resulting substring (10), not present in **V**, is therefore added to the vocabulary: **V**={1,0,01,10}. As the algorithm proceeds, the next two bits S(7)=0 and S(8)=1 are parsed. As the resulting substring 01 is part of **V**, S(9)=1 is appended to it, rendering 011. That value is added to the vocabulary, yielding **V**={1,0,01,10, 011}. The size of the vocabulary **V** is taken to be the complexity measure $c(n=9)=5$, which is also the number of steps required to construct **V**.

Lempel and Ziv also showed that the complexity *c(n)* of a binary random sequence where both symbols 0,1 have probability p(0) = p(1) = 0.5 is in the limit equal to $b(n) = n/\log_2(n)$. The magnitude *b(n)* gives, therefore, the asymptotic value of *c(n)*. By dividing *c(n)* by *b(n)* we get the normalized Lempel Ziv complexity measure *C(n)* which does not depend on the length *n* of the sequence.

$$C(n) = \frac{c(n)}{b(n)} \tag{1}$$

The normalized *C(n)* represents the rate of new substring occurrences in the sequence. *C(n)* values go from close to zero (for deterministic/ periodic sequences) to one (for totally random sequences). For example, the deterministic sequence s = 00000000000000000000000000000 can be parsed into 2 words {0, 0000000000000000000000000000}, given that the sequence can be reconstructed by inserting the first zero, and then repeatedly copying it to complete the sequence. The resulting LZ measure is $c(30) = 2$ and the normalized $C(30) = 0.335$. It should be noted that the normalized LZ measure is much more meaningful for larger values of *n* (for example, for s = 400x0 (400 zeros), $C(1000) = 0.02$) [19]. An example with varying values of *n* is provided in the following paragraphs. Figure 2 shows the algorithm, in pseudocode, for computing *C(n)*.

```
Function Normalized_LZC(s)
%-------------------------------------------------------------
------------
% This function takes the binary sequence s of size n as
argument and
% computes the normalized Lempel Ziv complexity measure
C(n)= c(n)/b(n)
%-------------------------------------------------------------
------------
 n=length(s);
 c=1; j=1; i=0;
 k=1; kmax=1; stop=false
 while not stop
   if (s[i+k] != s[j+k]) then
        if k>kmax then kmax=k

        i=i+1

        if (i EQ j) then
           c=c+1
           j=j+kmax
           if j+1>n then
                stop=1
           else
                i=0
                k=1
                kmax=1
           endif
        else
            k=1
        endif
   else
        k=k+1
        if j+k>n then
            c=c+1
            stop=1
        endif
   endif
 end while
 b=n/log2(n);
 Normalized_LZC = c / b;
 return Normalized_LZC);
```

Figure 2. Pseudocode of the algorithm to compute the LZ normalized complexity *C*(n) of a binary sequence of size n (adapted from Kaspar 1987)

The Lempel and Ziv complexity measure has been used successfully in a variety of domains. *LZ* complexity and derived LZ algorithms [40, 41] underlie the whole field of dictionary based lossless compression, and has triggered a huge amount of research, algorithms and application software (most desktop general purpose compression software such as PKZIP and WINZIP are implementations of dictionary based compression). In recent years, LZ has been widely used in biomedical applications to

estimate the complexity of discrete-time signals for recognition of structural regularities and for complexity characterization of DNA sequences [2, 12].

In this exploratory research work, we propose the use of the Lempel and Ziv normalized complexity measure to determine the degree of randomness (order / disorder) in the distribution of errors in a database file. The proposed procedure is simple: following Redman's criterion, equate each field of each record in the file to a cell, assign a zero to each errorless cell, and assign a 1 to each cell containing at least one error[4] [29, 30]. The next step is to concatenate all the cells to create a binary sequence s(*n*). Finally, compute the normalized LZ complexity C(*n*) of the sequence s(*n*) to quantify the degree of randomness of the errors in the file.

The LZ metric has several advantages. First, it is simple and fast to calculate for moderate sequence sizes. Second, it renders a number between 0 and 1 and in such way it provides an intuitive measure of the degree of randomness of the data errors. Third, it is well founded in theory and has plenty of practical applications (e.g. an alternative approach could compute the compression ratio between the generated sequence of binary digits and a totally random sequence of the same bit-length to determine the degree of randomness[5]). Fourth, it is one of the tests specified by The Computer Security Division of the National Institute of Standards and Technology (NIST) in its recommended battery of tests to assess the quality of random number generators, a key component of all modern cryptographic systems [31].

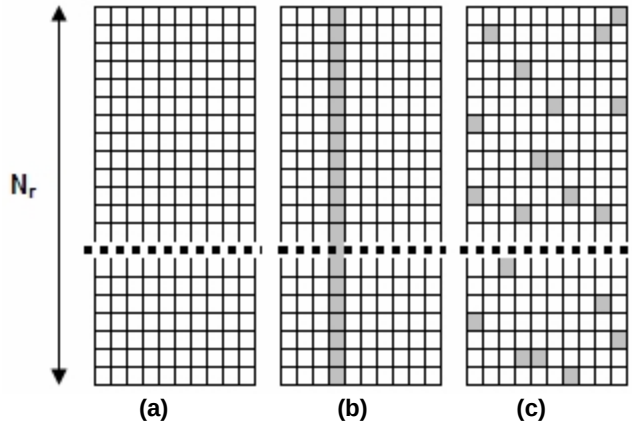

Figure 3. Impact of distribution of errors on the LZ measure

To further investigate the application of the LZ measure let us consider the example in Figure 3, using hypothetical files of 10 fields and N_r records. Case (a) has no errors; case (b) has a systematic error in the fourth field which gives way to 10% of errors in the file; case (c) has 10% of randomly distributed errors. Following the proposed procedure, in each case the records in each file are concatenated, replacing each field with a 0 if the field contains no errors, and with a 1 if the instance of the field is erroneous. This renders three sequences (S_a, S_b and S_c) of length $n = (10 \times N_r)$ bits. We ran a simulation with varying values of *n*. The results are shown in Figure 4. For S_a, and S_b, representing uniform / periodic patterns, *c(n)* remains constant and the normalized LZ measure C(*n*) starts with small values and rapidly tends to

[4] We understand that this is a simplified approach. Further elaboration may be required to deal with multiple errors per field.

[5] We performed some simple tests on binary files of standardized size (1000 rows, 50 columns, 2MB) A file with one column (1000 cells) filled with 1s was compressed to 2KB. A file with one row (50 cells) filled with 1s was compressed to 2KB. A file with 2100 1s randomly distributed across the 1000 x 50 matrix was compressed to 16KB.

zero. For S_c, c(n) grows steadily, and C(n) remains at a high value (around 0.5, out of an asymptotic maximum of 1). This could be taken as a signature of the high level of randomness of the file.

			No errors (all 0s)		All errors in 4th column		10% random errors	
n	log10(n)	b	*c(n)*	*C(n)*	*c(n)*	*C(n)*	*c(n)*	*C(n)*
1000	3	100	2	0.01993	4	0.03986	52	0.518221
10000	4	753	2	0.00265	4	0.00532	346	0.459755
100000	5	6021	2	0.00332	4	0.00066	2757	0.457928
1000000	6	50172	2	0.00004	4	0.00008	22869	0.455828

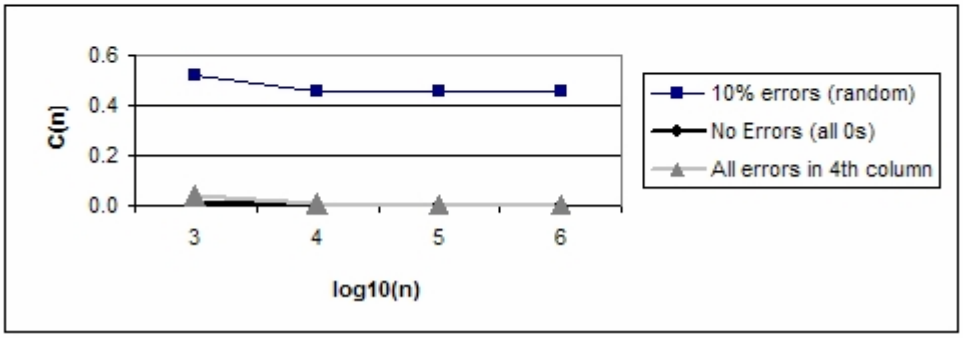

Figure 4. LZ normalized complexity measure C(n) as a function of $\log_{10}n$

POISSON DISTRIBUTION

Traditionally the Poisson distribution has been used to study arrival rates for queues. "The Poisson distribution is a discrete probability distribution that applies to occurrences of some event over a specified interval" ([35]; page 210). The Poisson distribution is regularly used in the evaluation of workload for online telecommunications systems. Analysts study the arrival rates per unit of time such as how many transactions entered the queue every second. Examples include Internet users logging on to web sites [35], patients arriving at clinics in 15 minute intervals [20], the number of phone calls received at a helpdesk [36] or a switchboard [24], the number of customers entering a gift shop per hour [36], and cars pulling into a gas station [35].

There are many examples that revolve around space rather than time. Examples include the number of bomb hits per a predefined geographic region in London during WW2 [35], the number of defects in the paint job on a new car [36], defects per foot of wire [24], the number of rivets on an airplane wing [34] and the number of typographical errors in a book [36].

These examples of number of defects in physical spaces led to our consideration of using the Poisson to help answer managerial questions related to the database quality. Bomb hits can be considered data errors and one can consider the region to be a database record (row) containing sub-regions (fields, attributes, cells). In this light, errors per row can be counted. One can start with a record being equal to a region and then determine how many hits (errors) there are per record. Another example which seems to come closest to the number of errors in a database is the number of typographical errors per page(s) of a book [36].

The calculation of the basic Poisson probabilities is straightforward and answers the general question, "What is the probability of x occurrences of an event in a given period or space?" There are several corollaries to this question which will be demonstrated. Importantly, it must be determined that the errors are randomly distributed, independent and equally distributed over the database. The KZ test

described above can be utilized. If the KZ test indicates non-randomness then there must be systematic patterns and the Poisson will not be used. The Poisson formula and brief examples follow.

$P(x) = (e^{-\lambda} * \lambda^{x}) / x!$ where λ = average number of events in the sample, x = the number of events being examined and e = the natural log at 2.71828 (or simply 2.72).

Suppose a store averages 2 customers every minute ($\lambda = 2$) and management wants to determine the probability of 3 customers arriving in any given minute. The probability that x = 3 is 18% as follows: $P(x=3) = (e^{-2} * 2^{3})/3! = .18$. [34].

Some of the questions that management might ask include:

1. What is the probability of zero events in any given interval?
2. What is the probability of less than x events for any given interval?
3. What is the most likely number of events for any given interval?

For illustration of answering those three questions follow an example from Pfaffenberger and Peterson [24] that gives a 10 minute interval with $\lambda = 1$ for a mean of 1 phone call per every 10 minutes. We apply the formula $P(x) = (e^{-\lambda} * \lambda^{x}) / x!$ to answer the given questions.

1. $P(x=0) = P(0) = 1/(e * 0!) = .3679$ or 37%
2. For less than x calls we use P(y where y = x-1), thus if we started with x = 4 then we calculate $P(x \leq 3) = P(0) + P(1) + P(2) + P(3) = .3679 + .3679 + .1839 + .0613 = 98\%$
3. To determine the most likely we simply observe the values that received the highest probabilities and discover that we have a bi-modal distribution with 0 and 1 being equal at .3679.

One more step is necessary to apply the Poisson as a database accuracy measurement. Suppose one would like to consider an interval with subintervals to account for records (rows) with fields (attributes). One can use the familiar λ to be the average number of events per unit of space or time (fields) and t = the number of units of space or time over which one counts the number of events occurring.

The previous example illustrated time intervals. The next examples involve units of geographical space. During World War II, London was assaulted with German flying-bombs on V-2 rockets. The British were interested in whether or not the Germans could actually target their bomb hits or were limited to random hits with their flying-bombs [8, 39]. The British mapped off the central 24 km by 24 km region of London into 1/2 km by 1/2 km square areas. Then they recorded the number of bomb hits, noting their location. This data is presented in the following table:

#bomb hits (k)	0	1	2	3	4	5
#areas with (k) bomb hits	229	211	93	35	7	1

To determine the average number of bomb hits per 1/2 km by 1/2 km section in London one computes the following weighted average: Σ (k hits * # areas with k hits) / N. Total bomb hits = (0*229 + 1*211 + 2*93 + 3*35 + 4*7 + 5*1) = (211 + 186 + 105 + 28 + 5) = 535. The total possible areas = (229 + 211 + 93 + 35 + 7 + 1) = 576. The average hits per area = (# of Hits) / (# possible areas) = 535 / 576 = 0.928819. Thus on average 0.928819 bombs hit a 1/2 km by 1/2 km section.

The probability of k bombs striking a given section is given by the Poisson distribution formula. $P(k) = (e^{-\lambda} * \lambda^k) / k!$ where λ = average number of events in the sample, k = the number of events being examined and e = the natural log at 2.71828 (or simply 2.72).
$P(k) = (e^{-.93} * .93^k) / k! = (2.72^{-.93} * .93^k) / k!$ A table is computed of theoretical values and these values are compared with the observed data to see if the bombs fell according to the Poisson.

#bomb hits (k)	0	1	2	3	4	5
Poisson Predictions for #areas with k bomb hits	227.5	211	98	30	7	1
Actual or observed #areas with k bomb hits	229	211	93	35	7	1

It would appear that the theoretical data and the observed data are almost identical. Thus, it appears that the number of hits over a geographical area follows the Poisson distribution[6]. The English correctly concluded that the hits were random. Now this logic is transformed into applying the Poisson as a metric for errors on a page. Given the random nature of the errors as determined by the LZ metric, one can consider the expected number of typographical errors on a page to be similar to the number of errors in a database, where pages are equivalent to rows.

Suppose the expected number of typographical errors on a page of a certain magazine is 0.2. The same management questions as above can be answered. For example, a) What is the probability that a page contains no errors? b) What is the probability that a page contains 2 or more errors? Apply the Poisson formula, $P(k) = (e^{-\lambda} * \lambda^k) / k!$ with $\lambda = 0.2$.
a) For zero errors, k = 0 and the formula condenses to $(e^{-.2} * .2^0) / 0! = e^{-.2} = 1/e^{.2} = .82$.
b) For 2 or more errors we calculate the probabilities of 0 errors and 1 error then subtract this result from 1. P>= 2 = 1- (P<=(1)) = 1 – (.0.982) = .018.

For a similar problem suppose there are 200 typographical errors in a 500 page document and one wants to find the probability that a given page has exactly 3 errors. First determine that $\lambda = 200/500 = .4$ per page. Set k = 3 and then $P(k) = (e^{-\lambda} * \lambda^k) / k! = (e^{-.4} * .4^3) / 3! = 0.007$ meaning that there is less than 1% chance that a page will have exactly 3 errors.

Finally, suppose there is a file with 100 records and the probability of any record having an error is .05. If one wants to find the probability that 0 through 5 records [or <= 5] will have an error, then use the Poisson with $\lambda = np = 100 * .05 = 5$. Then insert 5 for k in the formula $= (e^{-5} * 5^{5)} / 5! = .1755$.

A table can be built for this distribution that answers the question as to what is the probability of having less than or equal to a certain number of errors, p (a record having <= k errors), in a record as follows:

k =	0	1	2	3	4	5
$\lambda = 5$	.0067	.0337	.0842	.1404	.1755	.1755

Note that once λ is known there are comprehensive tables available for the Poisson distribution so the managerial questions may be readily answered [24].

CONCLUDING REMARKS AND FUTURE RESEARCH

[6] Note that the Chi-Square goodness of fit test may be used to verify the theoretical predictions with the observed results.

We have shown that there are potential deficiencies with the current accuracy metric for judging quality of databases. The issues raised at the start of this paper were that simple counts (#good, total, etc.) and ratios of cells (good/total) did not adequately distinguish the relative quality of a database as compared to other databases or to itself at a different point in time. This was shown in our comparisons of Figure 1 (a, b, c) above. While those were dramatically different in complexity, the accuracy or error rate measurements resulted in equal values. When one says that a database has a 10% error rate the current accuracy metric does not inform as to the severity of the quality of the database. Those deficiencies inhibit an ability to monitor for trends to determine if a database's quality is deteriorating or improving, to compare the quality of one database to another, to determine best-of-breed and benchmarking to determine if a project to improve quality did in fact improve the quality. The lack of a comprehensive metric also inhibits common actions such as negotiating the price, value and usefulness of data with an information vendor and qualifying data before it is applied from one database to another such as a data warehouse [23].

The suggestion of adding the LZ complexity measure significantly informs on the complexity (randomness) of the database quality. There are a number of issues, though, that deserve further consideration and might be the subject of future research.

- The measure yields a result between 0 and 1, with 0 (or close to 0) representing a deterministic pattern (systematic error), and 1 qualifying total randomness (white noise). However, the meaning of the intermediate values is not immediately evident from a quantitative perspective. In our example of Figure 4, case (c) had a fixed number (10%) of errors, randomly distributed in the file. The LZ value (close to 0.5) is large enough to identify a high level of randomness in the file but it does not give us a clear-cut complexity /randomness scale. Further studies are required to analyze this feature in more detail.
- As mentioned before, the LZ complexity measure is easily computable for moderate values of n but the LZ parsing algorithm, as defined by its authors [19], has a time complexity of $O(n^2)$, making it impractical for very large files. A simple solution is to divide the file into m segments of moderate size $n_{1=} n/m$, calculate $C(n_1)$ for each segment and compute an average value of $C(n_1)$. A similar approach is followed in the analysis of temporal bio-signals, where the LZ complexity is calculated over a window of time. Likewise, the LZ77 and LZ78 compression algorithms [40, 41] circumvent the $O(n^2)$ problem by restricting its search to a smaller window rather than the entire string.

In addition, once the randomness/complexity check is performed, the Poisson distribution may be used to answer a variety of managerial questions about the errors in the database. Tables of Poisson probabilities such as those found in Pfaffenberger may be used to help answer the questions ([24] p. 1152).

Further empirical research is needed to determine the efficacy of Poisson distribution in the field of data quality. While we demonstrated that it can nicely answer various management questions, we made a few assumptions in our work thus far. For example, what is the implication of a single cell having multiple errors and are those errors dependent or independent? Further work is needed to consider weighting of fields and severities of errors in all three of our accuracy-metric parameters. Finally, no attempt was made to include usefulness or usage as a factor in our accuracy-metric. Research could help determine if usage could be incorporated with weighting. While the field of data quality has been studying accuracy for many years there is a lot more to be done.

REFERENCES

1. *The Merriam-Webster Dictionary*, ed. H.B. Woolf. 1974, New York, NY: Pocket Books. 848.
2. Aboy, M., R. Hornero, R. Abasalo, and R. Alvarez, *Interpretation of the Lempel-Ziv Complexity Measure in the Context of Biomedical Signal Analysis.* IEEE Transactions on Biomedical Engineering, 2006. 53(11): p. 2282-2288.

3. Audini, B., A. Pearce, and P. Lelliott, *Accuracy, completeness and relevance of Department of Health returns on provision of mental health residential accommodation: A data quality audit.* Journal of Mental Health, 2000. 9(4): p. 365-370.
4. Ballou, D.P. and H.L. Pazer, *Designing Information Systems to Optimize the Accuracy-Timeliness Tradeoff.* Information Systems Research, 1995. 6(1): p. 51-72.
5. Ballou, D.P. and G.K. Tayi, *Enhancing Data Quality in Data Warehouse Environments.* Communications of the ACM, 1999. 42(1): p. 73-78.
6. Cappiello, C., C. Francalanci, and B. Pernici, *Time-related factors of data quality in multichannel information systems.* Journal of Management Information Systems, 2004. 20(3): p. 71-91.
7. Chaitin, G.J., *Randomness and mathematical proof.* Scientific American, 1975. 232(5): p. 47-52.
8. Clarke, R.D., *An Application of the Poisson Distribution.* Journal of the Institute of Actuaries, 1946. 72.
9. Davis, A., *September 11 Watch List Acquires Life of Its Own*, in *Wall Street Journal*. 2002: New York.
10. Dietel, E.J., *Recordkeeping integrity: Assessing records' content after Enron.* Information Management Journal, 2003. 37(3): p. 43-51.
11. Falk, R. and C. Konold, *Making Sense of randomness: Implicit encoding as a bias for judgment.* Psychological Review, 1997. 104: p. 301-318.
12. Gusev, V.D., L.A. Nemytikova, and N.A. Chuzhanova, *On the complexity measures of genetic sequences.* Bioinformatics, 1999. 15(12): p. 994-999.
13. Huang, K.-T., Y.W. Lee, and R.Y. Wang, *Quality Information and Knowledge*. 1999, Englewood Cliffs, NJ: Prentice Hall. 209.
14. Kac, M., *What is random?* American Scientist, 1983. 71(4): p. 405-406.
15. Kaspar, F. and H.G. Shcuster, *Easily calculable mesure for the complexity of spatiotemporal patterns.* Physical Review A, 1987. 36(2): p. 842-848.
16. Klein, B.D., *User perception of data quality: Internet and traditional text sources.* Journal of Computer Information Systems, 2001. 41(4): p. 9-25.
17. Kolmogorov, A.N., *Three approaches to the quantitative definition of information.* Problems of Information Transmission, 1965. 1: p. 1-7.
18. Lee, Y.W., L.L. Pipino, J.D. Funk, and R.Y. Wang, *Journey to Data Quality*. 2006, Cambridge, MA: MIT Press.
19. Lempel, A. and J. Ziv, *On the complexity of finite sequences.* IEEE Transactions on Information Theory, 1976. 22(1): p. 75-81.
20. Levin, R.I., D.S. Rubin, and J.P. Stinson, *Quantitative Approaches to Management*. 1986, New York: McGraw-Hill. 795.
21. Li, M. and P. Vitanyi, *A Introduction to Kolmogorov Complexity and its Applications*. 1997, London: Springer Verlag.
22. Nelson, R., P. Todd, and B. Wixom, *Antecedents of Information and System Quality: An Empirical Examination Within the Context of Data Warehousing.* Journal of Management Information Systems, 2005. 21(4): p. 199 - 235.
23. Olson, J.E., *Data Quality: The Accuracy Dimension*. 2003, San Francisco, CA. 293.
24. Pfaffenberger, R.C. and J.H. Patterson, *Statistical Methods*. 1987, Homewood, IL: Irwin. 1246.
25. Pierce, E.M., *Modeling Database Error Rates.* Data Quality, 1997. 3(1).
26. Pipino, L.L., Y.W. Lee, and R.Y. Wang, *Data Quality Assessment.* Communications of the ACM, 2002. 45(4): p. 211 - 218.
27. Prins, H., H.A. Büller, and J.H.M. Zwetsloot-Schonk, *Effect of discharge letter-linked diagnosis registration on data quality.* International Journal for Quality in Health Care, 2000. 12(1): p. 47-57.
28. Ravichandran, T. and A. Rai, *Total Quality Management in Information Systems Development.* Journal of Management Information Systems, 2000. 16(3): p. 119-156.

29. Redman, T., *Measuring Data Accuracy*, in *Information Quality*, R.Y. Wang, et al., Editors. 2005, M. E. Sharpe: Armonk, NY. p. 265.
30. Redman, T.C., *The Impact of Poor Data Quality on the Typical Enterprise.* Communications of the ACM, 1998. 41(2): p. 79 - 82.
31. Rukhin, A.L., J. Soto, J. Nechvatal, M. Smid, M. Levenson, D. Banks, M. Vangel, S. Leigh, and S. Vo, *A Statistical Test Suite for the Validation of Cryptographical Random Number Generators*. 2000, National Institute of Standards and Technology: Gaithersburg, MD.
32. Shannon, C.E., *A mathematical theory of communication.* Bell System Technical Journal, 1948. 27: p. 379-423, 623-656.
33. Solomonov, R.J., *A formal theory of inductive inference, pat 1.* Information and Control, 1964. 7: p. 1-22.
34. Summers, D.C.S., *Quality*. 2006, Upper Saddle River, NJ: Pearson/Prentice Hall. 819.
35. Triola, M.F., *Elementary Statistics*. 8 ed. 2001, Boston, MA: Addison Wesley, Inc. 855.
36. Utts, J.M. and R.F. Heckard, *Statistical Ideas and Methods*. 2006, Belmont, CA: Thompson. 748.
37. Wand, Y. and R.Y. Wang, *Anchoring Data Quality Dimensions in Ontological Foundations.* Communications of the ACM, 1996. 39(11): p. 86 - 95.
38. Wang, R.Y. and D. Strong, *Beyond Accuracy: What Data Quality Means to Data Consumers.* Journal of Management Information Systems, 1996. 12(4): p. 5 - 34.
39. Winkel, B.J., *Bombs, Baseball, e and the Poisson Distribution*. 2007, Rose-Hulman Institute of Technology: Terre Haute IN. p. Statistics Lecture.
40. Ziv, J. and A. Lempel, *A Universal Algorithm for Sequential Data Compression.* IEEE Transactions on Information Theory, 1977. 23(3): p. 337-343.
41. Ziv, J. and A. Lempel, *Compression of Individual Sequences via Variable-Rate Coding.* IEEE Transactions on Information Theory, 1978. 24(5): p. 530-536.