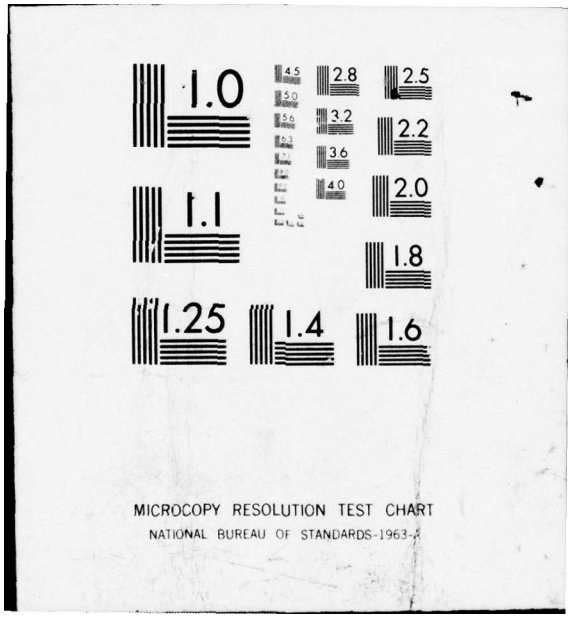
AD-A036 655 POLYTECHNIC INST OF NEW YORK BROOKLYN DEPT OF ELECTR--ETC F/G 9/2
SEEDING/TAGGING ESTIMATION OF SOFTWARE ERRORS: MODELS AND ESTIM--ETC(U)
JAN 77 B RUDNER F30602-74-C-0294

UNCLASSIFIED POLY-EE/EP-76-019 RADC-TR-77-15 NL

| OF |
AD
A036655

END

DATE
FILMED
3-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

RADC-TR-77-15
Technical Report
January 1977

SEEDING/TAGGING ESTIMATION OF SOFTWARE ERRORS: MODELS AND ESTIMATES

Polytechnic Institute of New York

Approved for public release;
distribution unlimited.

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public including foreign nations.

This report has been reviewed and is approved for publication.

APPROVED: Alan N. Sukert

ALAN N. SUKERT, Capt, USAF
Project Engineer

APPROVED: Alan R Barnum

ALAN R. BARNUM
Assistant Chief, Information Sciences Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: RADC-TR-77-15	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER: Technical
4. TITLE (and Subtitle): SEEDING/TAGGING ESTIMATION OF SOFTWARE ERRORS: MODELS AND ESTIMATES.	5. TYPE OF REPORT & PERIOD COVERED: Interim Report, 1 Apr 74 - 30 Jun 76
	6. PERFORMING ORG. REPORT NUMBER: Poly-EE/EP-76-019
7. AUTHOR(s): B. Rudner	8. CONTRACT OR GRANT NUMBER(s): F30602-74-C-0294
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Polytechnic Institute of New York, 333 Jay St, Brooklyn NY 11201	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F, 55500806
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (ISIS), Griffiss AFB NY 13441	12. REPORT DATE: January 1977
	13. NUMBER OF PAGES: 72
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same (12) 66p.	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(16) 5550 (17) 08

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Capt Alan N. Sukert (ISIS)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software Errors
Seeding/Tagging Estimates
Error Seeding
Error Tagging
Software Modeling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report concerns itself with seeding/tagging estimates of the number of software errors based on the number of errors either inserted deliberately in a program (seeded) or found by debugging (tagged), the number of errors found by a debugger unaware of the first set, and the number of errors appearing in both sets. Estimates from 3 models are discussed.

Model 1 assumes all errors are equally open to discovery at all times.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

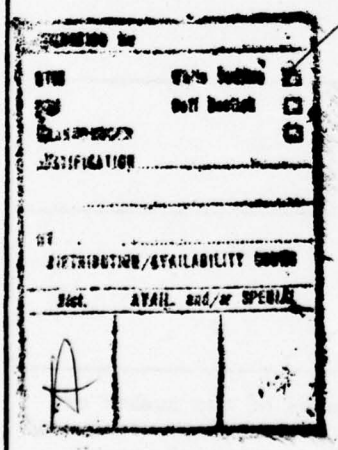
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408717

Model 2 and 3 assume categories of difficulty exist and that any error which appears can be assigned to the proper category. Model 2 does not assume the relative distribution of errors among categories is known, while Model 3 does.

The mean and mean-squared error of a maximum likelihood estimate and a modified maximum likelihood estimate are given for all 3 models. It is shown how these quantities vary with certain relations among the total number of errors, size of tagged or seeded set, and size of accompanying sample set. A procedure for determining optimum values for size of tagged or seeded set and number found by the second debugger is outlined. Finally, multi-trial estimates for parameters are found and compared with single-trial estimates.



Abstract

Seeding/tagging estimates of the number of software errors are computed from s,t and c where: t is the number of errors either inserted deliberately in a program (seeded) or found by debugging (tagged); s is the number found by a debugger unaware of the contents of the first set; and c is the number appearing in both sets.

Two types of questions can be raised. One type relates to the method and procedure: the introduction of new errors, the changing of a program by debugging, etc. The other relates to possible estimates, and their evaluation and comparison. This report concerns itself with questions of the second type. Estimates based on 3 models are discussed. The models are defined by assumptions regarding the equal or unequal difficulty of uncovering individual errors. Model 1 assumes all errors equally open to discovery at all times. Models 2 and 3 assume that categories of difficulty exist and that any error which appears can be assigned to the proper category. Model 2 does not assume that the relative distribution of errors in a program among categories is known, but Model 3 does. Estimates for Models 2 and 3 are shown to be closely related to those for Model 1.

The mean and mean-squared error of a maximum-likelihood estimate and a modified maximum likelihood estimate are given. It is shown how these quantities vary with certain relations among the total number of errors, size of tagged or seeded set and size of accompanying sample set. Curves are drawn which can be used to determine optimum values for s and t and a procedure is outlined for doing so.

More precise estimates can be obtained with several trials rather than one as described above. Several such estimates are examined and

discussed.

It is concluded in general terms that a reasonable investment of time will produce adequate estimates.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

SEEDING/TAGGING ESTIMATION OF SOFTWARE ERRORS: MODELS AND ESTIMATES

1.0 Introduction

1.1 Tagging/seeding census methods

It has been suggested that tagging/seeding census methods, used for many years to estimate the size of animal and fish populations, be borrowed from the arsenal of wildlife specialists for the purpose of estimating the number of bugs in a computer program. The initial seeding suggestion came from H. D. Mills*; the tagging alternative was proposed by M. Hyman*.

The "tagging" and "seeding" labels are descriptive of two ways in which the marked individuals required by the process are introduced into the population. In the tagging variant, also called a capture-recapture census, a sample of the population is captured, tagged and returned; a second sample, presumably containing some tagged individuals, is then captured. Under certain assumptions one can estimate the total population from the number of animals in each of the two captures, and the number recaptured, i.e., common to both. Seeding differs only in that the initial capture and release are replaced by the procedure of adding other marked individuals to the original population. If a uniform population is assumed, the two processes are statistically identical: the estimates used for the total original population in the tagging version are used for the augmented population in the seeding version. It is only necessary, in the latter case, to subtract the number of seeded individuals from the result. If uniformity

---

* In internal memoranda.

is not assumed, differences may arise; these are discussed in Chapter 8 in the context of application to computer programs.

1.2 Application to software errors

We can describe a process analogous to the foregoing animal census method for estimating the number of errors in a computer program at any point of its debugging life beyond the initial phase of correcting compiler-discovered errors. Suppose we give the program to two people to debug (or to continue debugging) independently, arranging that there be no contact between them. Each person tabulates and corrects errors as they appear. After an arbitrary period of time – which may differ for the two debuggers – we look at the results, i.e., the two sets of tabulated errors. Some errors will occur on both lists and some on only one. Consider one set to correspond to the animals first captured, tagged and released, and the other set to correspond to the second capture. The errors common to both lists correspond to the tagged animals included in the second capture. What we have described is a tagging analogue. For the seeding variant we would eliminate one debugger and instead insert an arbitrary number of known errors into the program. How many errors are seeded and which they are is not known to the remaining debugger. Most of this report is written with the tagging case in mind; however, translation to the seeding case is direct, in the manner described in Section 1.1.

It is hardly necessary to say that the tagging/seeding application to software errors raises more questions than can be answered readily. (Some, in fact, apply with equal validity to the original wildlife census process and have provoked many long discourses in the statistical journals.) For example, new errors may be introduced in the course of correcting

those found. Also two debuggers may correct errors differently, leading eventually to two quite different programs and different error counts.

We will, however, neither answer nor in fact examine all the questions one might ask. Instead we will start with the rash assumption that basically the process works. Our objective is to describe various models and to evaluate a collection of estimates which they support. If the approach described is feasible at all, one wants to know how good the results are likely to be, which estimates produce the most accurate and precise results, and under what conditions.

A subsequent report will describe an experiment which should reveal how well the technique works, what the problems are, perhaps what the answers to some of the questions are, and of course how our estimates compare if, indeed, it is possible to make them.

It will be assumed here that no new errors are introduced and that *different debuggers do not change the program in different ways in correcting* the same error.

1.3 Notation

The following symbols will be used uniformly:

- N = total number of errors initially present in a computer program (i.e., present when the test is begun). N includes seeded errors if the seeding variant is used.
- $\hat{N}$ = any estimate of N
- t = number of tagged errors; these are all the errors discovered by the first debugger (the tagger) in the tagging case, or the number of seeded errors in the seeding case.

s = number of sampled errors; these are all the errors discovered by the second debugger (the sampler) in the tagging case; or by the only debugger in the seeding case.

c = number of errors common to the tagged and sampled sets; i.e., the number found by both debuggers in the tagging case, or the number of seeded errors found by the sole debugger in the seeding case.

$\bar{c} = \frac{1}{n} \sum_{i=1}^{n} c_i$ = average of n values of c

$V(\hat{N}) = E[(\hat{N}-N)^2]$ = mean-squared variation of a biased estimate $\hat{N}$ about the true value N, as distinguished from $\text{var}(N) = E\{[\hat{N} - E(\hat{N})]^2\}$

$\sigma_e(\hat{N}) = [V(\hat{N})]^{\frac{1}{2}}$

$b(\hat{N})$ = bias of estimate $\hat{N} = E(\hat{N}) - N$

$[x]$ = greatest integer $\leq x$

$P_o \equiv P(0)$ = probability that $c=0$

Estimates

$\tilde{N}$ = ad hoc estimate

$N_o = \frac{st}{c}$

$N_1 = \frac{(s+1)(t+1)}{c+1} - 1$

N_{oi} = the i^{th} of several estimates of the form of N_o

N_{1i} = the i^{th} of several estimates of the form of N_1

$$\overline{N}_o = \frac{1}{n} \sum_{i=1}^{n} N_{oi} = \text{average of n estimates of the form of } N_o$$

$$\overline{N}_1 = \frac{1}{n} \sum_{i=1}^{n} N_{1i} = \text{average of n estimates of the form of } N_1$$

$$\tilde{N}_o = \frac{st}{\overline{c}}$$

$$\tilde{N}_1 = \frac{(s+1)(t+1)}{\overline{c}+1} - 1$$

2.0 Model 1 - Equal Probability Assumption

2.1 Description

The simplest model on which to base an estimate is that which assumes debugging to be completely random: that is, errors are said to be indistinguishable, each being found with probability 1/N where N = number of errors in the program at the time.

2.2 Discussion

The basic assumption that all errors have equal probability of discovery may not reflect the facts of life in computerland. Although there is little hard data, the general impression is that some errors are easy to find and would quickly be turned up by any debugger while some consistently resist descovery.

One can readily conceive of several factors which may make for variable difficulty. For example: type of instruction; particular test data; debugger technique; location (beginning or end of the program, within a loop, hidden by other errors, etc.).

Individually some such factors would cause underestimation by an estimate based on the equal probability assumption and some would cause overestimation. As an example of the latter, variation due to debugger technique would tend to make the overlap, c -- which appears in the denominator of estimates -- too small because the tagger and sampler are, so to speak, fishing in different waters. On the other hand, a variation more closely related to the nature of the error itself would cause an underestimation since the sampler would in a practical sense have available only the easier bugs and the estimate would actually be of that subset.

It is not known whether some particular factors of this nature have an

overriding effect, or whether all would be well enough served by throwing them into the statistical mash of equal probability.

3.0 Ad Hoc Estimate $\tilde{N}$

A reasonable estimate is suggested directly by the equal probability assumption. We have a program with N bugs initially; let, for example, the unknown number N be 100. If the first debugger, the tagger, finds t = 20 bugs, t/N or 1/5 of the total errors are tagged. The second debugger, the sampler, finds s = 25 bugs. If the tagged and untagged bugs can be found with equal ease (or difficulty), both should appear in the s-element sample in about the same proportion as in the entire set of errors: $t/N \simeq c/s$, where c is the number of tagged bugs turning up in the sample.

Since t, s and c are known, N is approximately determined by the ratio. This is our first estimate: $\tilde{N} = \frac{st}{c}$ (or the nearest integer to st/c). Suppose the 25 sampled bugs included 6 tagged bugs. Then $\tilde{N} = \frac{25 \times 20}{6} = 83$. c would have to be 5 to make the ratio exactly true and the estimate exactly right.

In the seeding version, the original errors, N_x, would have numbered 80, t = 20 would have been seeded; $N = N_x + t = 100$ would have been estimated by $\tilde{N} = 83$ as above, and the estimate of the original number of errors would have been $\tilde{N}_x = \tilde{N} - t = 63$.

The example also illustrates a concomitant of all estimates considered, which we will call integer error: c = 4, 5, 6 give respectively $\tilde{N}$ = 125, 100, 83; no in-between values are possible. Clearly the integer constraint on c can cause a large error in the estimate to arise from a small -- even the smallest possible -- deviation in c from its "ideal" value. Integer error will be discussed again in Section 6.

4.0 Maximum likelihood estimate N_o

4.1 Distribution of data values

For a more formal derivation of estimates, we recognize that the tagging/seeding procedures outlined describe a standard experiment in sampling without replacement [1]. The collection of N errors is analogous to an urn of balls, identical except for color: t balls -- the tagged errors -- are red while the N-t remaining are white. The debugging experiment is equivalent to having a blindfolded sampler reach in and withdraw s balls. Some balls in the sampled group will be red and some white. The number of red balls sampled is a discrete random variable $\underset{\sim}{c}$ which can assume only integral, non-negative values. The probability that $\underset{\sim}{c}^*$ will have some particular value c is given by the hypergeometric distribution:

$$P(c \mid s, t, N) = \frac{\binom{t}{c}\binom{N-t}{s-c}}{\binom{N}{s}} \equiv \frac{\binom{s}{c}\binom{N-s}{t-c}}{\binom{N}{t}} \equiv \frac{s!\,t!\,(N-s)!(N-t)!}{N!\,c!\,(s-c)!(t-c)!\,(N-s-t+c)!}$$

The mean and variance of the distribution are respectively [2]:

$$E(c \mid s, t, N) = \frac{st}{N}$$

$$\text{var}(c \mid s, t, N) = \frac{st}{N} \cdot \frac{(N-s)(N-t)}{N(N-1)}$$

The distribution is symmetrical with respect to s and t, implying reasonably enough, that it does not matter which debugger we call the sampler and which the tagger, nor whether s or t is larger.

A lower bound on c is certainly 0. However, if $s + t > N$, small positive values are impossible. For example, if 55 of a total of 100 bugs are tagged, and the sampler finds 50, there must be an overlap of at least

---

* The tilde will generally be omitted in this report; it should be clear from the context whether the random variable or a particular value is intended.

5 in the tagged and sampled sets; therefore $P(c \mid 50, 55, 100) = 0$ for $c=0, 1, \ldots, 4$. The distribution as given does not exist for those values since the factorial of a negative number is undefined; however, if we replace the factorials by the corresponding gamma functions, we do get $P=0$ for $0 \leq c < 5$.

Since the number of common bugs can exceed neither the number tagged nor the number sampled, the limits of c are described by

$$\max (0,\ s + t - N) \leq c \leq \min (s, t).$$

Figure 1 shows the distribution for different values of s, t and N, continuous lines replacing point probabilities for readability.

4.2 Maximum Likelihood Estimate

In the case at hand s and t are known parameters, c is experimentally determined, and the problem is to estimate the unknown parameter N. The maximum likelihood estimate N_o is shown in [1] to be

$$N_o = \left[\frac{st}{c}\right]$$

N_o is essentially equal to the ad hoc estimate $\tilde{N}$. Since $\frac{st}{c}$ does not exist for $c=0$, we arbitrarily define N_o to be $2st$ when $c=0$. It is a reasonable choice since it amounts to replacing $c=0$ by $c=1/2$.

The properties of N_o are derived and examined in Chapman [3]. It is shown there that N_o is a consistent* but positively biased estimate, the bias and variance decreasing with increasing $\frac{st}{N}$, i.e., with increasing mean of the distribution. Because of the bias, the mean-squared error $V(\hat{N}) = E[(\hat{N}-N)^2]$ rather than the variance was taken as a measure of dispersion. Both bias and mean-squared error have rather unwieldy expressions

---

* Consistency here means that the estimate approaches N in probability in either of two circumstances: either (1) N increases while s/N and t/N remain constant or (2) N remains constant and the product st increases.

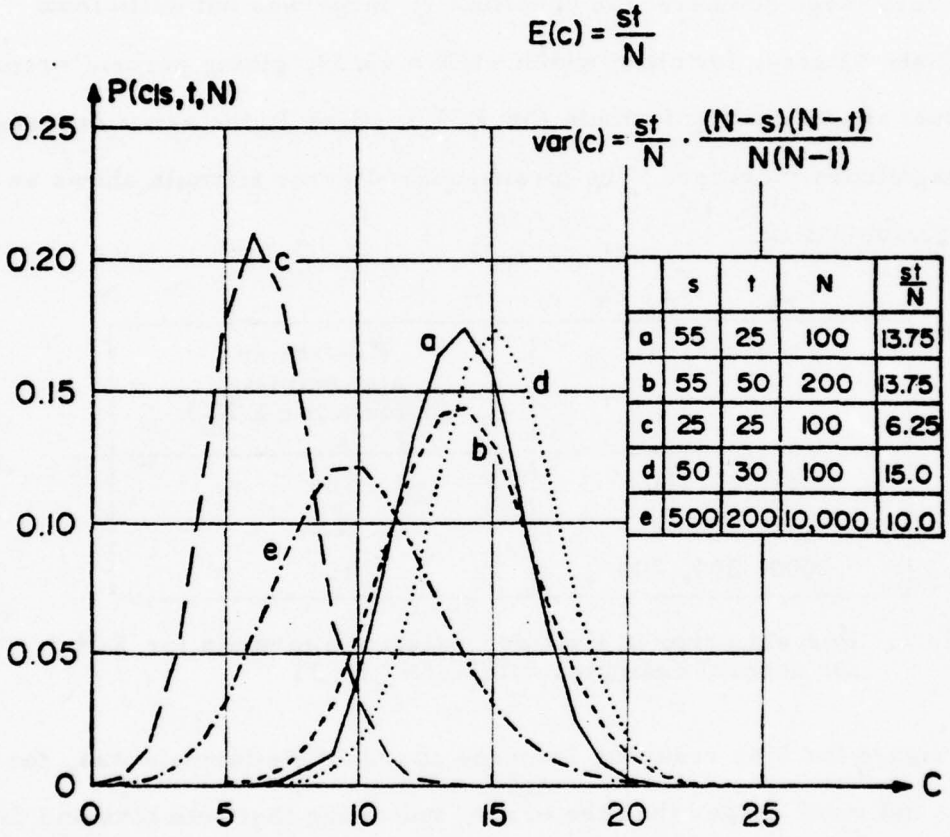


FIG. 1. HYPERGEOMETRIC DISTRIBUTION FOR VARIOUS PARAMETER VALUES.

and Chapman gives much simpler approximate forms, derived for $st/N \geq 10$.

The work covered in this report had been substantially completed when inconsistencies in certain results* led to a check of the simplified formulas for bias and dispersion. It was discovered that certain approximations used in deriving the formulas introduced serious error unless N, s and t were actually very large numbers; the condition $\frac{st}{N}$ large was not sufficient. Table 1 lists 3 cases, for all of which $st/N = 13.33$, giving percent error in the original approximation formula for $E(N_o)$. Clearly the error decreases as the magnitudes increase. The mean-squared error formula shows an even larger deviation.

$(st/N = 13.3)$ N, s, t	% error in approximate formula for $E(N_o)$
30, 20, 20	7.9
270, 60, 60	3.5
3000, 200, 200	1.1

Table 1. Percent error in first approximation formula for $E(N_o)$ for several examples with $st/N = 13.33$

The figure for bias resulting from the approximate formula was, for small N, not much larger than the error, indicating that both bias and dispersion might actually be considerably lower than appeared. Consequently,

* There were actually 2 sources of inconsistency. Since Chapman's approach did not apply in the multi-trial case (Sec. 6) another approach was used and a new formula derived which, with the number of trials reduced to one, should have given about the same result as was obtained with Chapman's formula. However, the figures for dispersion in the example tested were far different. The second inconsistency was noted when comparison was made with some specific cases in a tabulation containing means and variances computed directly from probabilities [4].

it was necessary to derive second-order approximations for bias and mean-squared error which would be more accurate than Chapman's first-order approximations but still simpler than the exact expressions. Such approximations would permit quick calculation, provide insight into the manner in which bias and dispersion change with changing parameter values, and facilitate comparison with other estimates.

Two sets of bias and mean-squared error formulas were obtained, one using the method applied by Chapman but eliminating the offending approximations, and the other based on a Taylor's series expansion of $\frac{st}{c}$. The first derivation is described in Appendix 1 and the second in Appendix 2. The Taylor's series approach was initially applied to find the mean and dispersion of estimates based on several data values (see Sec. 6), a problem to which the Chapman method is not applicable. Although such was not their raison d'être, the resulting formulas can be used to verify the calculations for N_o as well.

On the basis of the new approximations, additional interesting information was obtained on the manner in which bias and mean-squared error change with the parameters, information which would be useful in designing an actual estimation effort.

4.3 Bias

The new approximation for the expected value of N_o derived from Chapman's exact result is

$$E(N_o) \simeq st\,[\alpha_1 + \alpha_2 + 2\alpha_3 + 6\alpha_4 + \ldots + (m-1)!\,\alpha_m] \qquad (1)$$

where

$$\alpha_1 = \frac{N+1}{(s+1)(t+1)}$$

$$\alpha_i = \alpha_{i-1}\,\frac{N+i}{(s+i)(t+i)}, \quad i = 2, 3, \ldots$$

The requirements for accuracy (see Appendix 1) are the following:

1. Enough terms must be included in the sum, which is a truncated version of an infinite sum, to leave the remainder insignificant. Four or five terms have been found sufficient.

2. The probability that c=0 must be very small. By referring to the examples of hypergeometric distribution in Fig. 1, one sees that this occurs when the peak is far from 0, i.e., when the mean of the distribution, st/N, is large. In fact, common sense tells us that large samples are almost certain to have elements in common; i.e., $P(0) \simeq 0$. $st/N \geq 3$ seems to be sufficient for accuracy unless N is very large (in which case the variance of the distribution and therefore P_o is large).

An alternative form of Eq. (1), derived by simple manipulations (see Appendix 1) is:

$$E(N_o) \simeq N[k_1 + k_2 (\frac{N}{st}) + 2k_3 (\frac{N}{st})^2 + \ldots + (m-1)!k_m(\frac{N}{st})^{m-1}] \qquad (1a)$$

$$\text{where } k_1 = \frac{1 + 1/N}{(1 + 1/s)(1 + 1/t)}$$

$$k_i = k_{i-1} \frac{1 + i/N}{(1 + i/s)(1 + i/t)}, \quad i = 2, 3, \ldots$$

The quantities k_i are close to 1 and increase to 1 as a limit as s, t and N increase. If we set all $k_i=1$, we arrive at Chapman's approximate formula

$$E(N_o) \simeq N [1 + (\frac{N}{st}) + 2 (\frac{N}{st})^2 + \ldots] .$$

A method described in [5] for deriving the expected value of a function of a random variable by means of a Taylor's series expansion was applied (see Appendix 2) leading to

$$E(N_o) \simeq N [1 + q (\frac{N}{st}) + 3 q^2 (\frac{N}{st})^2] \qquad (2)$$

$$\text{where } q = \frac{(N-s)(N-t)}{N^2}$$

This is subject to the same caveat as Eq. (1): truncation effect and $P_o \neq 0$ are possible sources of error. Both tend to show up for small $\frac{st}{N}$, and for large values of N, s, t, i.e., values for which $\min(s,t) > > \frac{st}{N}$.

The bias, b, of an estimate $\hat{N}$ is defined by $E(\hat{N}) = N + b$. The quantity of greatest interest is the ratio $\frac{b}{N}$ (or percent bias = $\frac{b}{N} \times 100\%$) since to estimate $N = 100$ as $\hat{N} = 120$ is clearly a grosser error than to estimate $N = 1000$ as 1020.

The percent bias of N_o varies in 3 different ways: (1) with size of tagged and sampled sets relative to total number of errors, quantified by the ratio $\frac{st}{N}$; (2) with the total number of errors N; and (3) with size of sampled set relative to size of tagged set, $\frac{s}{t}$. The nature of each variation, with the other 2 sources held constant, is considered next.

1. $\frac{b}{N}$ decreases as $\frac{st}{N}$ increases, for N and $\frac{s}{t}$ constant.

The consistency of the estimate, shown by Chapman, implies that this is so in the limit. For *finite* N, *s* and *t*, Eq. (1a) shows that variation in $E(N_o)$, with N constant, depends principally on the $\frac{N}{st}$ factors. While the k_i factors increase with increasing s and t, all are less than and close to 1 and vary very little over large changes in s and t. (See Fig. 2(a)).

2. $\frac{b}{N}$ increases with N for $\frac{st}{N}$ and $\frac{s}{t}$ fixed.

In this case, the only variation in $E(N_o)$ is with the quantities k_i (see Eq. (1a)) which increase with increasing N under the given conditions. The common upper limit of the k_i's is 1 which occurs only for infinite s, t and N. Chapman's formula, which results if all $k_i = 1$, therefore gives an upper limit to the bias ratio, holding for very large N. (See Fig. 2(b)).

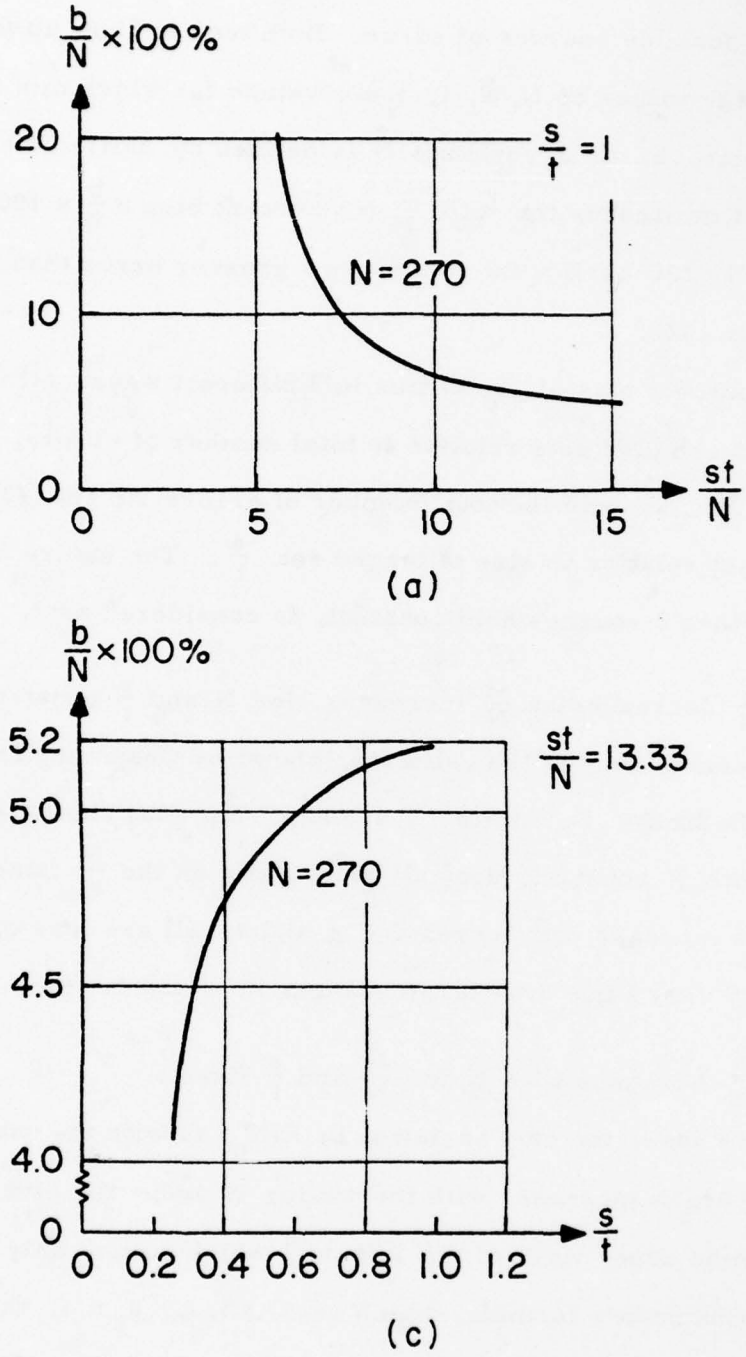


FIG. 2. VARIATION WITH SEVERAL PARAMETER RELATIONS OF PERCENT BIAS OF ESTIMATE N_o.

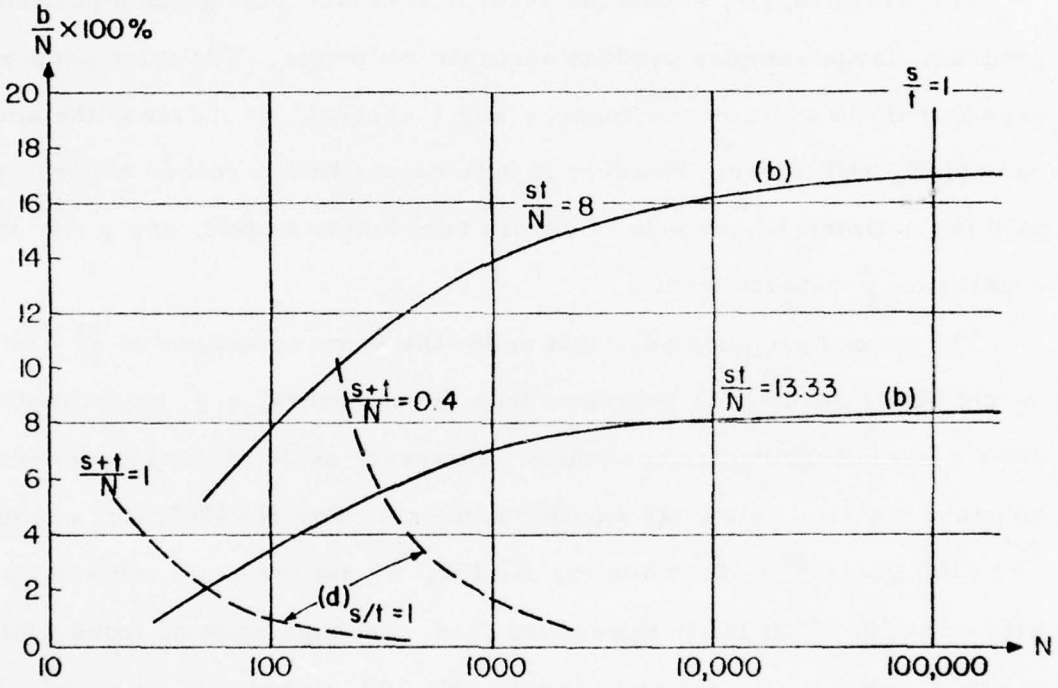


FIG. 2. (B) AND (D) . (Continued)

3. For $\frac{st}{N}$ and N fixed, $\frac{b}{N}$ is greatest when $\frac{s}{t} = 1$.

For $\frac{st}{N}$ and N fixed, the product st is fixed, and $\frac{s}{t} = 1$ implies $s = t = \sqrt{st}$. We can show (see Appendix 4) that

$$\frac{1 + i/N}{(1 + i/s)(1 + i/t)} \leq \frac{1 + i/N}{(1 + i/\sqrt{st})^2}$$

from which it follows that k_i, and therefore b/N, are maximum for $s = t$. (See Fig. 2(c).)

The first property states the unsurprising fact that, given a particular program, large samples produce accurate estimates. The third property says that if, in addition, we make s and t unequal, we increase the accuracy of N_o still more. However in both cases, the increased accuracy is paid for in time: larger sets of errors take longer to find, and $s + t$ increases as $\frac{s}{t}$ departs from 1.

The second property says that under the same conditions of $\frac{st}{N}$ and $\frac{s}{t}$ we get better results for programs with fewer errors, e.g. by estimating N after some debugging has been done. However, as N increases, keeping $\frac{st}{N}$ constant requires relatively smaller samples. For $N = 1000$, for example, $s = t = 100$ gives $\frac{st}{N} = 10$, while for $N = 250$, we get the same value of $\frac{st}{N}$ with $s = t = 50$. That is, in the second case, 100 bugs must be found while in the first, with N four times as large, only 200, or twice as many bugs must be found. If we spend the same time relative to N and find 400 bugs in the first case, we increase $\frac{st}{N}$ by a factor of 2 and decrease bias considerably. To sum up the argument, if we keep the debugging time, as measured by $s + t$, proportional to N, then N_o has smaller bias for large N. (See Fig. (2d).)

4.4 Mean-squared Error

New approximate formulas for $V(N_o)$, the variation about the true value N, were derived using Chapman's method (Appendix 1) and the Taylor's series method (Appendix 2). They are respectively,

$$V(N_o) \simeq N^2 \left\{ 1 + \frac{st}{N} \left[-2\alpha_1 + \left(\frac{st}{N} - 2\right)\alpha_2 + \left(\frac{st}{N} - 4\right)\alpha_3 + \ldots \right.\right. \tag{3}$$

$$\left.\left. \ldots + \left\{ A_{m-1} \frac{st}{N} - 2\,(m-1)! \right\} \alpha_m \right] \right\}$$

where the α's are defined as in Eq. (1)

and

$$A_{m-1} \triangleq (m-1)! \sum_{j=1}^{m-1} \frac{1}{j}$$

$$V(N_o) \simeq N^2 \left[q\,(N/st) + 9q^2\,(N/st)^2 \right] \tag{4}$$

where q is defined as in Eq. (2).

An alternative form for Eq. (3) is

$$V(N_o) \simeq N^2 \left[(1 + k_2 - 2k_1) + (3k_3 - 2k_2) \frac{N}{st} + (11k_4 - 4k_3) \left(\frac{N}{st}\right)^2 + \ldots \right. \tag{3a}$$

$$\left. + \left(A_{m-1} k_m - 2(m-2)!\, k_{m-1}\right) \left(\frac{N}{st}\right)^{m-2} \right]$$

where the k's are defined as in Eq. (1a).

The formulas hold under the same conditions as the mean formulas: $P_o \simeq 0$, and low truncation error. Furthermore the same generalizations can be made with respect to the variation of $V(N_o)$ with N, s, t.

5.0 Modified Maximum Likelihood Estimate N_1

5.1 Bias and Mean-squared Error

An intermediate result in Chapman's derivation of $E(N_o)$,

$$E\left(\frac{1}{c+1}\right) = \frac{N+1}{(s+1)(t+1)}(1-K), \quad \text{where } K = \begin{cases} \frac{N-s-t}{N+1} P_o & \text{for } s+t \leq N \\ 0 & \text{otherwise} \end{cases}$$

suggests the modified estimate

$$N_1 = \frac{(s+1)(t+1)}{c+1} - 1$$

as a means of reducing the bias to practically zero assuming $P_o \simeq 0$. For

$$E(N_1) = (s+1)(t+1)\,E\left(\frac{1}{c+1}\right) - 1 = (N+1)(1-K) - 1$$

$$E(N_1) = N - K(N+1) \quad \text{where } K \simeq 0 \text{ if } P_o \simeq 0$$

$$\therefore \quad E(N_1) \simeq N$$

The bias is negative but very small even for small $\frac{st}{N}$. Consider for example, the case $N=6$, $s=2$, $t=3$, with $\frac{st}{N} = 1$. $E(N_1)$, computed exactly, is 5.8, and b/N is 3.3% whereas $E(N_o)$ is 6.6 with b/N = 10%.

However, for N_1 as for N_o, b/N increases if st/N is held fixed but N increases. If N=20, s=4, t=5, $\frac{st}{N}$ is still 1 but $E(N_1)$ is now 16.9 and b/N = 15.5%.

An additional advantage of N_1 is the fact that its variation about N is somewhat lower than $V(N_o)$ for N greater than about 50. Below 50, $V(N_o)$ is smaller. The second-order approximation for $V(N_1)$, under the same approximation rules as $E(N_o)$ and $V(N_o)$ is (see Appendix 1)

$$V(N_1) \simeq (s+1)^2(t+1)^2[\alpha_2 + \alpha_3 + 2\alpha_4 + 6\alpha_5 + \ldots + (m-2)!\,\alpha_m] - (N+1)^2 \qquad (5)$$

or

$$V(N_1) \simeq (s+1)^2 (t+1)^2 \left(\frac{N}{st}\right)^2 \left[(k_2 - k_1^2) + k_3 \left(\frac{N}{st}\right) + 2k_4 \left(\frac{N}{st}\right)^2 + \ldots + (m-2)!\, k_m \left(\frac{N}{st}\right)^{m-2} \right] \quad (5a)$$

Some comparative figures for N_o and N_1 are shown in Table 2 and in Table 3 of Section 6.3. The variation of $\sigma_e(N_1)/N$ with relations among the parameters, as described in detail in Sec. 4.3, is plotted in Fig. 3.

5.2 Useful Range

It is obviously possible to make accurate and precise estimates with large enough samples; the limiting case of $s = t = N$ produces a perfect estimate. Whether a good estimate can be made with considerably smaller samples is the issue. N_1 has almost no bias so the major problem resides in the variance (which, for zero bias, equals the mean-squared error). As Eq. (5) and Fig. 3(a) show, the variance is low for the ratio st/N large enough. But large ratios can be attained with relatively small samples only for N large. For N = 3000, for example, st/N = 13.33 can be realized with s = t = 200, or one-fifteenth of N; but for N = 30, st/N = 13.33 requires s = t = 20, two-thirds of N. Fortunately, Fig. 3(b) shows that smaller values of st/N are required to give a specified value of σ_e/N at the 30-error level than at 3000. The $\frac{s+t}{N} = 1.0$ curve in Fig. 3(d) shows the minimum value of σ_e/N which can be attained if we limit s and t to half of N. If we are willing to accept larger samples, we can, of course, do better for the smaller values. Larger samples mean more time. For the same time relative to N, estimates of larger programs will have lower σ_e/N (Fig. 3(d)). Curves such as those of Fig. 3 can be exploited to design an estimation test with knowledge of the trade-off between time and precision.

$N = 270 \quad \frac{st}{N} = 13.33$

$s = 60$

$t = 60$

C	N_o	N_1
9	400	371
10	360	337
11	327	309
12	299	285
13	276	265
14	257	247
15	239	232
16	224	218
17	211	206
18	199	195

Table 2. Comparison of N_o and N_1 for different experimental results: - one example.

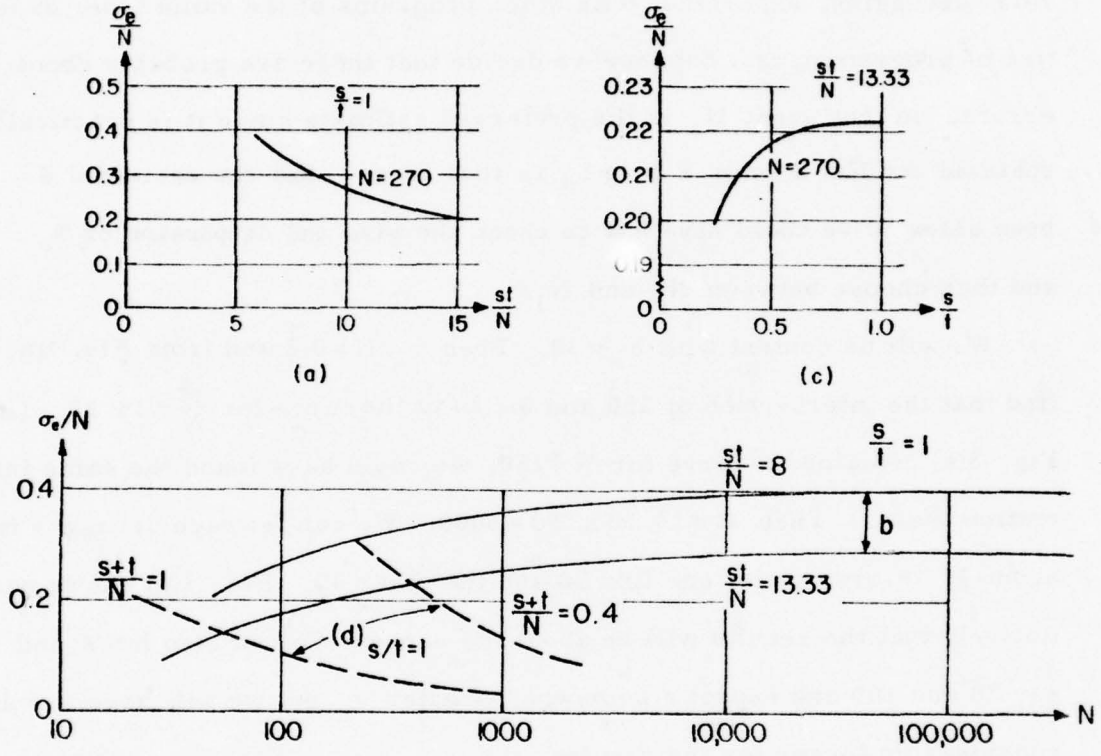


FIG. 3. VARIATION WITH SEVERAL PARAMETER RELATIONS OF σ_e/N FOR ESTIMATE N_1.

5.3 Design of a Seeding/Tagging Reliability Test

The procedure is very simple. Our objective is to pick values for s and t which will be likely to produce an estimate of the quality we want. We begin with a ballpark estimate of the number of errors in the program, based on whatever information we have – length of program, amount of previous debugging, experience with other programs of the same type, expertise of programmers. Suppose we decide that there are probably about 150 errors. In that event N_1 is the preferred estimate since it is practically unbiased and has a lower V than N_o in that range. Had the estimated N been below 50 we would have had to check the bias and dispersion of N_o and then choose between N_o and N_1.

We will be content with $\sigma_e = 30$. Then $\sigma_e/N = 0.2$ and from Fig. 3(b) we find that the intersection of 150 and 0.2 is on the curve for $\frac{st}{N} = 13.33$. (If Fig. 3(a) contained a curve for N = 150, we could have found the same information there.) Then $st = 13.33 \times 150 = 2000$. We can let each debugger find about 45 errors, or let one find 50 and the other 40. Fig. 3(c) shows qualitatively that the results will be about the same. We can also let s and t be, say 20 and 100 and expect a somewhat smaller σ_e but we will have to wait considerably longer for the results.

The cost beyond that for the debugging which would have to be done anyway would be identical for all choices since the additional cost is only for the common bugs and the expected number of those is $\frac{st}{N} = 13.33$.

The situation would be a little different if the program were not to be completely debugged. The test could, for example, be a means of comparing different programming techniques. In that case, it would not only take longer but would also be more expensive to find 120 bugs than to find 90.

6.0 Multi-Trial Estimates

6.1 Advantages

We have up to now been discussing an estimation process involving two debuggers. Suppose we use 3 or more and consider the output of each pair to be a separate result; m debuggers will give $n = \frac{m(m-1)}{2}$ possible data values which can be combined to provide a new estimate with the following possible advantages:

1. reduced integer error
2. reduced variance, or
3. smaller samples and less debugging time for the same variance.

6.1.1 Integer Error

It was noted in Section 3 that in any seeding/tagging calculation an error arises from the fact that c is an integer, assuming one of only $\min(s,t)+1$ values. Any estimate, it follows, must also have one of only so many values despite the fact that N may actually be an integer from $\max(s,t)$ to infinity. This is particularly bothersome when the numbers are relatively small. With a population of, say, 10^4 and $s = t = 1000$, data values $c = 100$ and $c = 99$ lead to maximum likelihood estimates of 10,000 and 10,101 respectively, a difference of only 1% of the true population. But with a population of 100 and $s = t = 25$, data values $c = 6$ and 7 respectively provide estimates of 104 and 89, a span of 15%, with no possibility that any value obtained with the given estimate will fall within the range. The difference at the center of the distribution is of the order of $\frac{N}{st} \times 100\%$ of N. Values of c further away from the mean will lead to even larger separations; $c = 4$ and 5 for example give $N_o = 156$ and 125 respectively, a difference of

31% of N. These are not improbable values; 4 is about 1 standard deviation away from the mean of the distribution.

Integer error is automatically reduced when several data values are used, whether the averaging is done on the data values themselves, or on the several estimates derived from the individual data values. In the last example, for instance, using 4.5, the average of $c = 4$ and $c = 5$, in the maximum likelihood formula gives an estimate of 138, while averaging the values of N_o obtained with $c = 4$ and 5 gives $N_o = 140$. Either way, values are possible which cannot be obtained from a single trial, and increasing the number of trials increases the number of new values. More in-between values are likely to occur if the final estimates rather than the data are combined – the average of $c = 4$ and 6 is not a new value, for example, while the nonlinearity of the estimates makes repeats when estimates are averaged improbable – but such estimates may be less desirable for other reasons.

6.1.2 Variance and Mean-squared Error

We can reasonably expect some reduction in variance in a multi-trial process regardless of the estimate formula. However, the bias of the new estimate as well as the degree of improvement in variance do depend on the formula.

One combining mode is to compute an estimate for each data value, using any single-trial formula, and average the resulting n single-trial estimates. Then $\text{var(average)} = \frac{1}{n}\text{var(each)}$*. However $V = \text{var} + (\text{bias})^2$.**

* If m debuggers are used, the $n = m(m-1)/2$ possible data values are not statistically independent and the variance relationship is not exactly true. However, to avoid complications assume the n values are approximately independent, or consider that n truly independent tests are made with 2n debuggers.

** See Appendix 4

Therefore the reduction in V for a multi-trial estimate found in this way depends on the change, if any, in bias as well as in variance.

Alternatively, we can average the n values of c getting $\bar{c} = \sum_{i=1}^{n} c_i$, and replace c by $\bar{c}$ in any single-trial estimate formula. Although $V(\bar{c}) = \text{var}(\bar{c}) = \frac{1}{n} \text{var}(c)$, the effect of the replacement is not obvious and must be examined anew for each formula.

Taking another point of view, we can trade reduced variance for a quicker estimate by reducing s and t. Since V varies more or less inversely with the product st (Eq. (3a)), having tagged and sampled sets of size $\frac{\overline{t}}{n}$ and $\frac{\overline{s}}{n}$ respectively for an n-trial estimate will keep the variance of the n-trial estimate approximately equal to that of the corresponding single-trial estimate using sets of size s and t. Smaller samples mean less time. The time saving may be more than proportional to the reduction in s and t since errors probably becomes progressively harder to find as the total number remaining decreases. That is, it takes longer for one debugger to find 50 errors than for 2 to find 25 each, starting with the program in the same state. However, choosing time-saving in preference to reduced variance would, if the estimate is biased, increase the bias which also varies as $\frac{1}{st}$ (Eq. (1a)).

It might also be borne in mind in contemplating multi-trial estimating that the multiple debugging is not all wasted; each debugger added to the process finds errors others do not find, thereby contributing to the necessary over-all debugging of the program.

6.2 Averaging Single-trial Estimates: $\overline{N}_o$ and $\overline{N}_1$

Let the estimate N_o be the average of n maximum-likelihood estimates

associated with n independent experimental values c_i, i = 1, 2, ..., n:

$$\overline{N}_o = \frac{1}{n} \sum_{i=1}^{n} N_{oi} = \frac{1}{n} \sum_{i=1}^{n} \frac{st}{c_i} .$$

Although the variance would be reduced by the expected factor of $\frac{1}{n}$, the bias would remain unchanged: $E(\overline{N}_o) = E(N_o)$, so V would be reduced by less than the variance.

$$V(\overline{N}_o) = \text{var } (\overline{N}_o) + b^2$$

$$= \frac{1}{n} \text{ var } (N_o) + b^2$$

$$= \frac{1}{n} [V(N_o) - b^2] + b^2$$

$$= \frac{1}{n} [V(N_o) + (n - 1) b^2]$$

$$= \frac{1}{n} V(N_o) + \frac{n-1}{n} b^2$$

If the bias is large, therefore, $V(\overline{N}_o)$ may be considerably greater than $\frac{1}{n} V(N_o)$. In view of the almost zero bias of N_1 for $\frac{st}{n} \geq 3$ and N not too large, it would make more sense to average N_1:

$$\overline{N}_1 = \frac{1}{n} \sum_{i-1}^{n} N_{1i}$$

Here,

$$E(\overline{N}_1) = E(N_1) \simeq N$$

$$V(\overline{N}_1) \simeq \frac{1}{n} V(N_1)$$

6.3 Averaging Data Values: $\overline{\overline{N}}_o$ and $\overline{\overline{N}}_1$

Define estimate $\overline{\overline{N}}_o$ by

$$\overline{\overline{N}}_o = \frac{st}{\bar{c}} \quad \text{where} \quad \bar{c} = \frac{1}{n} \sum_{i=1}^{n} c_i$$

Since a single s appears in the formula for $\overline{\overline{N}}_o$, all samples must have the same size. This implies that $s = t$; $\overline{\overline{N}}_o$ can therefore be written $s^2/\bar{c}$.

The bias and variance of $\overline{N}_o$ and $\overline{N}_1$ were found directly from the bias and variance of N_o and N_1, the original computation of which was based on certain expansions of the reciprocal of a random variable with hypergeometric distribution. The same method cannot apply to $\overline{\overline{N}}_o$ because $\bar{c}$ is not hypergeometric.

However we can use the Taylor's series method mentioned in Section 4.3. The results (see Appendix 2) are

$$E(\overline{\overline{N}}_o) \simeq N \left[1 + \frac{q}{n} \left(\frac{N}{st}\right) + 3 \frac{q^2}{n^2} \left(\frac{N}{st}\right)^2 \right] \tag{6}$$

$$V(\overline{\overline{N}}_o) \simeq N^2 \left[\frac{q}{n} \left(\frac{N}{st}\right) + 9 \frac{q^2}{n^2} \left(\frac{N}{st}\right)^2 \right] \tag{7}$$

Setting $n = 1$ reduces $\overline{\overline{N}}_o$ to N_o. Because of the factors $\frac{1}{n}$ and $\frac{1}{n^2}$, $E(\overline{\overline{N}}_o) < E(N_o)$ which equals $E(\overline{N}_o)$. Similarly, because of the $\frac{1}{n^2}$ factor $V(\overline{\overline{N}}_o) < \frac{1}{n} V(N_o) < V(\overline{N}_o)$. The conclusion is that $\overline{\overline{N}}_o$ is a better estimate than $\overline{N}_o$, as the example in Table 3 shows.

Finally we mention the estimate $\overline{\overline{N}}_1$ having the form of N_1, but with $\bar{c}$ replacing c. Its bias and variance can be derived in the same way as those of $\overline{\overline{N}}_o$. Mean and mean-squared error formulas for the estimates considered are collected in Table 4.

Example

$N = 270$ $\qquad \frac{st}{N} = 13.33$

$s = 60$

$t = 60$ $\qquad n = 3$

	$\overline{\overline{N}}_o$	$\overline{N}_o$	N_o	N_o	N_1	$\overline{N}_1$	$\overline{N}_o$
E( )	274	284	284	284	270	270	284
V( )	1253	1683	4658	4519	3605	1202	1637
σ_e( )	35.4	41.0	68.3	67.2	60.0	34.7	40.5
	Taylor's series calculations			Non-Taylor's series calculations			

Table 3. Comparison of mean and dispersion of single- and multi-test estimates for one example.

Estimate	E ( )	V ( )
1. $N_o = \frac{st}{c}$	$st\left[\alpha_1+\alpha_2+2\alpha_3+\ldots+(m-1)!\alpha_m\right]$	$N^2\left\{1+\frac{st}{N}\left[-2\alpha_1+(\frac{st}{N}-2)\alpha_2+\ldots+\{A_{m-1}(\frac{st}{N})-2(m-1)!\}\alpha_m\right]\right\}$
2. $N_1 = \frac{(s+1)(t+1)}{c+1} - 1$	N	$(s+1)^2(t+1)^2\left[\alpha_2+\alpha_3+2\alpha_4+\ldots+(m-2)!\alpha_m\right]-(N+1)^2$
3. $\bar{N}_o = \frac{1}{n}\sum_{i=1}^{n} N_{oi}$	$E(N_o) \equiv N+b$	$\frac{1}{n}\left[V(N_o)+(n-1)b^2\right]$
4. $\bar{N}_1 = \frac{1}{n}\sum_i N_{1i}$	N	$\frac{1}{n}\,V(N_1)$
5. $\bar{\bar{N}}_o = \frac{st}{\frac{1}{n}\sum_i c_i}$	$N\left[1+\frac{q}{n}(\frac{N}{st})+3\frac{q^2}{n^2}(\frac{N}{st})^2\right]$	$N^2\left[\frac{q}{n}(\frac{N}{st})+9\frac{q^2}{n^2}(\frac{N}{st})^2\right]$
6. $\bar{\bar{N}}_1 = \frac{(s+1)(t+1)}{\frac{1}{n}\sum_i c_i+1} - 1$		

where $\alpha_1 = \frac{N+1}{(s+1)(t+1)}$

$\alpha_i = \alpha_{i-1}\,\frac{N+i}{(s+i)(t+i)}$, $i = 2, 3, \ldots$

$q = \frac{(N-s)(N-t)}{N^2}$

$A_{m-1} = (m-1)!\sum_{j=1}^{m-1} 1/j$

Table 4. Approximate formulas for means and mean-squared errors of estimates.

7.0 Confidence Intervals

Some measure of the dispersion of an estimate is necessary to provide information on the range within which, given the outcome of any particular trial, the true value of the quantity sought may be expected to lie. If the estimate is biased and has a large variance we have no great faith that the true value is close to the estimated value. Inserting the variance of the estimate in Chebyshev's inequality affords us one way of quantifying the spread. Another is available if we know the distribution of the estimate. Still another method, useful when the distribution and variance of the estimate are not known, involves the calculation of confidence intervals based only on the known distribution of data values, and on the particular value found experimentally.

Confidence limits a_1 and a_2 are two random functions of the estimate under study and of an arbitrary non-negative constant $\epsilon \leq 1$ for which we make the following claim: If the true value of the quantity being estimated is in the interval $[a_1, a_2]$, then the estimate actually computed, or else some value closer to the true value, would occur in $(1-\epsilon)100\%$ of trials made.

Each estimate is a function of one or more data values. Consequently any function of an estimate can be expressed as a function of the data variable c, or the set $\{c_i\}$ in a multi-trial procedure. The probability that a particular value of an estimate will occur is identical with the probability that the data points giving rise to that value will occur. Since the calculation of the confidence limits depends on the distribution of the data variable, the limits for all estimates depending on a single value are identical, and can be found by means of Eq. (9) in Section 7.1.

7.1 Confidence Limits for the Estimates N_o and N_1

For a 100ϵ% confidence level our first requirement is to find an interval enclosing a set of data values occurring in $(1-\epsilon)100\%$ of trials. One way to do this is to replace the hypergeometric distribution which describes the probability of c by its normal approximation (mean = st/N, variance $\sigma^2 = \frac{st}{N} \cdot \frac{(N-s)(N-t)}{N^2}$). This done, we determine λ such that $(1-\epsilon)100\%$ of all occurrences of c will be within a distance of $\lambda\sigma$ from the mean. λ is tabulated directly in [6] p. 558, or can be found using a table of error functions. An interval on the c-axis satisfying the stated condition is described by

$$P\left\{\frac{st}{N} - \lambda\sigma \leq c \leq \frac{st}{N} + \lambda\sigma\right\} \geq 1 - \epsilon \tag{8}$$

Our objective is to find the two values of N, N_a and N_b, for which the value of c found experimentally is at the ends of the allowed interval (see Figure 4). The left inequality should provide the largest mean = st/N for which c is still in the interval, and therefore the smallest N, i.e., the lower confidence limit. We might replace c by st/N_o or by $\frac{(s+1)(t+1)}{N_1 + 1} - 1$ depending on the estimate of interest; either N_o or N_1, computed from c, would then be the fixed quantity rather that c. The procedure and results would be identical, as noted previously, since we are still governed by the assumed normal distribution of c as expressed in (8) rather than by the distribution of N_o or N_1, neither of which is known.

If σ were constant we could immediately solve the two inequalities for N, thereby finding the confidence limits with no difficulty. Since σ is, instead, a function of N, the procedure is not quite so straight-forward. Details appear in Appendix 3 in which it is shown that the confidence limits

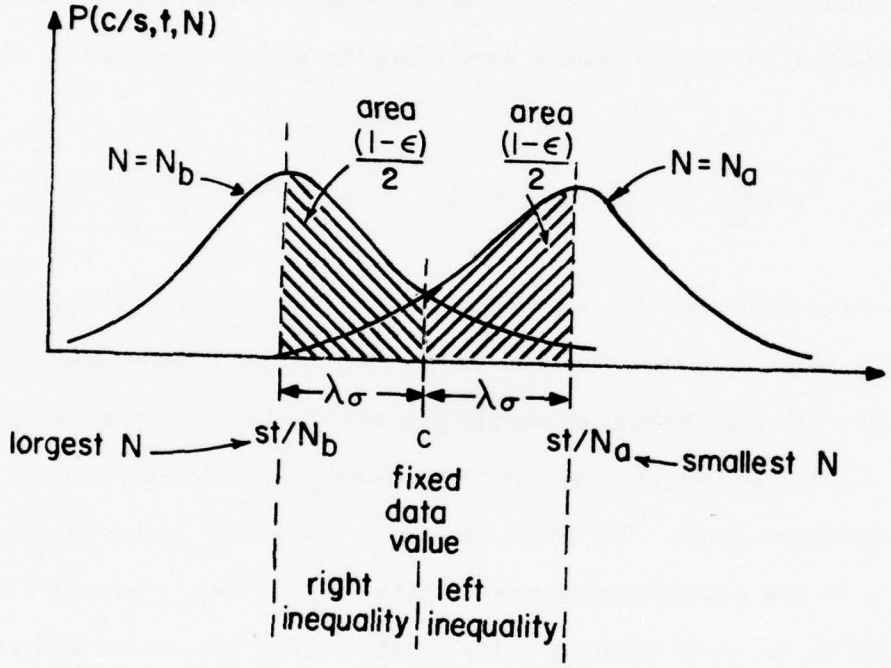


FIG. 4. CONFIDENCE INTERVAL.

at the 100ϵ % level are the 2 largest roots of $g(N) = 0$, where

$$g(N) = N^3 - \frac{st}{c}(2 + \frac{\lambda^2}{c})N^2 + \frac{st}{c^2}[st + \lambda^2(s+t)]N - (\frac{st}{c}\lambda)^2 , \qquad (9)$$

and λ depends on the preselected ϵ .

Examples:

$s = t = 25$

1) $c = 4$

$$N_o = \frac{st}{c} = 156$$

$$N_1 = \frac{(s+1)(t+1)}{c+1} - 1 = 134$$

10% confidence level: $\epsilon = 0.1$, $\lambda = 1.6449$

$g(N) = N^3 - 418N^2 + 29,699N = 66,064$

Confidence interval = [88, 328]

50% confidence level: $\epsilon = 0.5$, $\lambda = .6745$

$g(N) = N^3 - 330N^2 + 25,303N - 11,107$

Confidence interval = [121, 209]

2) $c = 7$

$N_o = 89$

$N_1 = 84$

10% confidence level:

$g(N) = N^3 - 213N^2 + 9701N - 21,575$

Confidence interval = [62, 148]

Confidence intervals are not unique. We can obtain limits more symmetrically placed about the estimate by choosing different values for the

left- and right-hand occurrences of λ in (8) and modifying the procedure accordingly. In any event, the results are approximate since they are based on the normal approximation to the distribution of c.

7.2 Confidence Limits for the Estimates $\overline{\overline{N}}_o$ and $\overline{\overline{N}}_1$

The random variable in the expressions for $\overline{\overline{N}}_o$ and $\overline{\overline{N}}_1$ is $\bar{c} = \frac{1}{n} \sum_{i=1}^{n} c_i$. $\bar{c}$ is an asymptotically normal random variable (central limit theorem) with mean equal to the mean of each individual c, or st/N; and variance equal to 1/n times the variance of each:

$$\bar{\sigma}^2 = \text{var}\,(\bar{c}) = \frac{st}{nN} \frac{(N-s)(N-t)}{N(N-1)}$$

Determining λ as before, we have

$$P\left\{\frac{st}{N} - \lambda\bar{\sigma} \leq \bar{c} \leq \frac{st}{N} + \lambda\bar{\sigma}\right\} \geq 1 - \epsilon$$

which leads us to another version of g(N):

$$g(N) = N^3 - \left[1 + \frac{st}{\bar{c}}\left(2 + \frac{\lambda^2}{n\bar{c}}\right)\right]N^2 + \frac{st}{\bar{c}}\left[2 + \frac{st}{\bar{c}} + \frac{\lambda^2(s+t)}{n\bar{c}}\right]N - \left(\frac{st}{\bar{c}}\right)^2\left(1 + \frac{\lambda^2}{n}\right) \qquad (10)$$

Example 3

$s = t = 25$

$c_i = 4,\ 11,\ 6\ ;\ \bar{c} = 7$

$\overline{\overline{N}}_o = 89$

For $\epsilon = 0.1$

$g(N) = N^3 - 192\,N^2 + 8727N - 15,165$

10% confidence limits are 70 and 120.

The width of the interval is considerably less for $\bar{\bar{N}}_o$ than for N_o computed with $c = 7$, as anticipated from its smaller mean-squared error. In fact, except for the effect of bias, the two widths should be proportional to the respective standard deviations, the ratio being $1/\sqrt{n}$. In this case it is almost exactly that, the bias apparently playing a small role.

If, instead of using the central limit theorem, we use the normal approximation for each c_i we have a slightly different variance, $\bar{\sigma}^2 = \frac{st}{nN} \frac{(N-s)(N-t)}{N^2}$, (see [1]) leading to slightly different results:

$$g(N) = N^3 - \frac{st}{\bar{c}}(2 + \frac{\lambda^2}{n\bar{c}})\, N^2 + \frac{st}{\bar{c}^2}[\, st + \frac{\lambda^2}{n}(s+t)\,]\, N - \frac{(st\lambda)^2}{n\bar{c}^2} \qquad (10a)$$

The same example now gives limits of 71 and 118, almost identical with the above.

8.0 Other Models - Assumption of Variable Intrinsic Difficulty

All the estimates which have been examined were based on the assumption of purely random choice: all errors were, one might say, laid out before any debugger who had but to close his eyes and choose. The discussion on equal probability in Section 2 noted several varieties of challenge which might be launched against that hypothesis. In this section we attempt to describe models providing for variable intrinsic difficulty.

8.1 Model 2 - Variable Difficulty, Program Distribution Unknown

Make the following assumptions:

1) All bugs can be assigned at sight to categories based on difficulty of discovery.

2) Within each category, errors are undifferentiated -- subject to random discovery with equal probability.

Suppose there are k difficulty categories. Tag (or seed) and sample as before. By virtue of assumption 2, $c_i/s_i = t_i/N_i$ where t_i, s_i, c_i are the tagged, sampled and common bugs respectively in the i^{th} category; some may be 0. N_i is the unknown number of program bugs in the i^{th} category. In principle, one may apply all Model 1 information to each category separately, deriving category estimates using any estimator previously discussed. For example, using N_o for simplicity, we have category estimates $N_{oi} = s_i t_i / c_i$, $i = 1, \ldots, k$, which can be found whenever s_i and t_i are non-zero.* Since the more difficult categories will probably be empty at first, we will not in general have an estimate of the total population. We can, in theory at least, continue to test until enough errors in all categories are

---

* We retain the convention that $N_{oi} = 2 s_i t_i$ when $c_i = 0$.

available. However, a possibly more efficient way is to estimate whatever categories appear in sufficient numbers after a brief test to make the estimate reliable; continue debugging without testing* (i.e., with one debugger) but keeping count of the number of bugs found in each category which has not yet been estimated; and finally conduct similar tests to estimate the missing categories when their appearance is frequent enough, adding the pre-estimate count in each case to get a number comparable with the initial estimates. If the reason for making the original estimate is to gauge reliability at the end of a finite debugging process, an error count -- though not by category -- would be required in any event in order to estimate how many remain still undiscovered.

As an example consider Table 5 where 3 categories of difficulty are assumed. The true figures in the total column are of course unknown. Two sets of experimental values for c_i are shown. In a) the c_i were chosen at their expected values for the true data $E(c_i) = s_i t_i / N_i$; the category estimates are therefore exactly right. In b) the values are not ideal. The r column and the remaining estimates will be defined in connection with Model 3.

A major difficulty is that numbers may be small; getting large enough samples within each category for low bias and variance may require extensive testing.

8.2 Model 3 - Variable Difficulty, Program Distribution Known

A third assumption makes it possible to complete the estimate with one trial, from incomplete category estimates:

3) The distribution ratio of program errors by category is known.

---

* This is suggested in order to avoid the cost of continued duplicate debugging.

Category		Tagged	Sampled		Total	c_i	
i	Type	t_i	s_i	r_i	N_i	a) Ideal	b) Non-ideal
1	easy	400	480	.6	1200	160	150
2	medium	60	100	.3	600	10	12
3	hard	40	0	.1	200	0	0
	total	500	580		2000	170	162

Model 2

a) $N_{o1} = \frac{480 \times 400}{160} = 1200$

$N_{o2} = \frac{100 \times 60}{10} = 600$

N_{o3} no estimate

b) $N_{o1} = \frac{480 \times 400}{150} = 1280$

$N_{o2} = \frac{100 \times 60}{12} = 500$

N_{o3} no estimate

Model 3 - first procedure

$N_o(r_1) = \frac{1200}{.6} = 2000$

$N_o(r_2) = \frac{600}{.3} = 2000$

Average $\overline{N}_o = 2000$

$N_o(r_1) = \frac{1280}{.6} = 2133$

$N_o(r_2) = \frac{500}{.3} = 1667$

Average $\overline{N}_o = 1900$

Table 5. Example with errors differentiated by difficulty (Models 2 and 3).

8.2.1 First Estimating Procedure - for Tagging or Seeding

The new assumption provides us with the ratio $r_i = N_i/N$ for all i.* Using any one of the category estimates of Section 8.1 we find $N_o = N_{oi}/r_i$. In fact we have as many estimates of N as we have category estimates. Table 5 contains an example of this estimating procedure.

N_o has the same ratio of bias and standard deviation to mean as N_{oi}:

$$E(N_o) = \frac{1}{r_i} E(N_{oi})$$

$$var(N_o) = \frac{1}{r_i^2} var(N_{oi}) \quad , \quad \sigma(N_o) = \frac{1}{r_i} \sigma(N_{oi})$$

$$V(N_o) = \frac{1}{r_i^2} V(N_{oi}) \quad , \quad \sigma_e(N_o) = \frac{1}{r_i} \sigma_e(N_{oi})$$

8.2.2 Second Estimating Procedure - For Seeding Only

With the third assumption we can also use the seeding variant to estimate N directly without finding category estimates. Let the program have $E = E_1 + \ldots + E_k$ errors, E_i representing the number in the i^{th} category. Construct and insert a matching set of $t = t_1 + \ldots + t_k$ errors, $t_i/t = E_i/E$, $t_i \neq 0$. The total number of errors after seeding is $N = N_1 + \ldots + N_k$

where $N_i = E_i + t_i$

and $\frac{N_i}{N} = \frac{E_i}{E} = \frac{t_i}{t}$ from the matching condition.

The debugger finds $s = s_1 + \ldots + s_k$ bugs, where s_i may be 0.

---

* If the seeding approach is used, the ratio r_i is computed with the seeded bugs included in N_i. The seeded bugs need not be distributed among the categories in the same proportion as the original program errors.

Since the seeded bugs are assumed indistinguishable from the original, we can again reasonably expect that

$$\frac{c_i}{s_i} = \frac{t_i}{N_i}$$

where now c_i and s_i may jointly be 0, but $t_i \neq 0$.

The total number of seeded bugs uncovered is

$$c = \sum_i c_i = \sum_i s_i \frac{t_i}{N_i} = \sum_i s_i \frac{t}{N} = \frac{st}{N}$$

We are therefore led to the same ad hoc estimate as for the equal probability case, $\hat{N} = \frac{st}{c}$.

The c_i are hypergeometric by virtue of assumption 2, but the distribution of their sum c is unknown, although asymptotically normal. The Taylor's series method (Appendix 2) with normal approximation for c_i (or only for c if k is large) can be used to find the mean and mean-squared error of the estimate $\frac{st}{c}$. Mean and variance of c are the sum of the means and variances of c_i:

$$E(c_i) = \frac{s_i t_i}{c_i} \qquad E(c) = \sum_{i=1}^{k} \frac{s_i t_i}{c_i} = \frac{st}{N}$$

$$\text{var}(c_i) = \frac{s_i t_i}{N_i} \cdot \frac{(N_i - s_i)(N_i - t_i)}{N_i^2} \qquad \text{var}(c) = \sum_{i=1}^{k} \text{var}(c_i)$$

Higher moments are found from the normal approximation

$$\mu_3(c) = 0$$

$$\mu_4(c) = 3\,[\text{var}(c)]^2$$

Table 6 shows an example of this approach with ideal and non-ideal experimental values. (Both are ideal in the sense that the seeded set matches the true distribution exactly.)

Category i	Type	Seeded t_i	Sampled s_i	Total N_i	c_i a) Ideal	c_i b) Non-ideal	
1	easy	60	480	1200	24	20	a) $\hat{N} = \frac{st}{c} = \frac{580 \times 100}{29}$
2	medium	30	100	600	5	6	$= 2000$
3	hard	10	0	200	0	0	b) $\hat{N} = \frac{580 \times 100}{26}$
	total	100	580	2000	29	26	$= 2231$

Table 6. Proportional seeding example with errors differentiated by difficulty (Model 3).

9.0 Conclusions

The modified maximum likelihood estimate considered under the equal-probability assumption is the estimate of choice in a single-trial test if the total number of errors exceeds about 50; its bias is practically zero and its variance reasonable. Its variance was found, furthermore, to vary in predictable ways with various ratios among error population, sample size and size of tagged or seeded set. As a consequence it is possible to design a seeding/tagging test optimally for the desired precision. Graphs make the choice of s and t a simple procedure. Estimates of larger values can be made in relatively less time.

For $N < 50$ a decision must be made between N_1, and N_o with its higher bias but lower mean-squared error.

Multi-trial procedures can decrease the dispersion still further. Of the two types considered, the better one replaces the random variable c by its average over the several trials.

A brief treatment of estimates for models other than equal probability indicates that the estimates are closely related to those for equal probability.

In summary, estimates of adequate accuracy and precision are available. The viability of seeding/tagging reliability tests rests on the answers to the practical questions which can be raised.

Appendix 1. Derivation of Second Order Approximations for $E(N_o)$, $V(N_o)$ and $V(N_1)$.

A. Derivation of Eq. (1)

The exact expression for $E(N_o)$ derived by Chapman consists essentially of the first m terms, m arbitrary, of an infinite series plus a remainder term. It can be written in the following form:

$$E(N_o) = st\{(1-K)[\alpha_1 + \frac{r_2}{r_1}\alpha_2 + 2\frac{r_3}{r_1}\alpha_3 + 6\frac{r_3}{r_1}\alpha_4 + \ldots + (m-1)!\frac{r_m}{r_1}\alpha_m]$$

$$+ \frac{P_o}{\alpha_1}[1 - \frac{r_2}{r_1}\alpha_2 - 2\frac{r_3}{r_1}\alpha_3 - \ldots - (m-1)!\frac{r_m}{r_1}\alpha_m]$$

$$+ (1-P_o)E(R_m \mid s \neq 0)\}$$

where $K = \begin{cases} \frac{N-s-t}{N+1} P_o & \text{for } s+t \leq N \\ 0 & \text{otherwise} \end{cases}$

$$\alpha_1 = \frac{N+1}{(s+1)(t+1)}$$

$$\alpha_i = \frac{N+i}{(s+i)(t+i)}\alpha_{i-1} \quad \text{for } i \geq 2$$

$$r_i = 1 - \sum_{c=0}^{i-1} P(c \mid N+i,\ s+i,\ t+i)$$

R_m = remainder term

To derive his first order approximation Chapman assumes that for $\frac{st}{N} \geq 10$, the following approximations hold:

$$P_o \simeq 0,\ K \simeq 0,\ R_m \simeq 0,\ \text{all } r_i \simeq 1,\ \text{all } \alpha_i \simeq \left(\frac{N}{st}\right)^i .$$

The result is

$$E(N_o) \simeq N \left[1 + \left(\frac{N}{st}\right) + 2 \left(\frac{N}{st}\right)^2\right]$$

from which it appears that the bias depends only on the ratio $\frac{N}{st}$, or $\frac{st}{N}$. The critical assumption is $\alpha_i \simeq \left(\frac{N}{st}\right)^i$, which is close for N, s and t very large but otherwise introduces considerable positive error, increasing as N gets smaller.

The remaining assumptions seem justified:

1. Since the series converges, $R_m \to 0$.

2. P_o is generally small except for N very large and $\frac{st}{N}$ very small simultaneously.

$\frac{N}{st}$ small means that the mean of the distribution of c is close to the origin, and N large means that for a given value of $\frac{st}{N}$ the variance is large since $\frac{(N-s)(N-t)}{N(N-1)} \to 1$. Large variance implies relatively high probability at $c = 0$. Some order-of-magnitude values are:

$\frac{st}{N}$	N	P_o
15	100	10^{-12}
13.75	200	10^{-8}
13.75	100	10^{-11}
13.33	27,000	10^{-6}
10	100	10^{-7}
6.25	100	10^{-4}
1	100,000	0.4
1	6	0.2

We can therefore assume, except where signalled by small $\frac{st}{N}$, that $P_o \simeq 0$.

3. It follows that K, which is less than P_o, is approximately 0 and $1-K \simeq 1$.

4. For i small the probabilities in the expressions for r_i are in the tail of the distribution, and are small for reasonable $\frac{st}{N}$ (see Figure 1). Therefore $r_i \simeq 1$ and $\frac{r_i}{r_1} \simeq 1$.

5. In most cases, the second term is very much less than the first term and can safely be ignored. The first term is dominated by α_1 for reasonable values of $\frac{st}{N}$ as the definition of α_1 shows. The ratio of the second term to the first is about $\frac{P_o}{\alpha_1^2}$. For $\frac{st}{N}$ exceedingly large $\alpha_1 \simeq \frac{N}{st}$ is very small and $\frac{P_o}{\alpha_1^2}$ can be significant. However, if $P_o \simeq 10^{-6}$ and $\frac{st}{N} \simeq 10$, $\frac{P_o}{\alpha_1^2} \simeq 10^{-4}$. That is, we make an error of about .01% in $E(N_o)$ by neglecting the second term.

If we make only the assumptions discussed above omitting the α_i assumption, we are left with

$$E(N_o) \simeq st\,[\alpha_1 + \alpha_2 + 2\alpha_3 + \ldots + (m-1)!\alpha_m]. \tag{1}$$

B. Derivation of Equation (1a)

Let $\alpha_o = \frac{N}{st}$. Define k_i for i = 1, 2, ... by $\alpha_i = k_i \alpha_o^{\,i}$.

Rewrite Eq. (1) as

$$E(N_o) \simeq N\left[k_1 + k_2\left(\frac{N}{st}\right) + 2k_3\left(\frac{N}{st}\right)^2 + \ldots + (m-1)!\,k_m\left(\frac{N}{st}\right)^{m-1}\right].$$

It remains to be shown that

$$k_1 = \frac{1 + 1/N}{(1 + 1/s)(1 + 1/t)}$$

which is done by factoring $\alpha_o = \frac{N}{st}$ from α_1; and that

$$k_i = k_{i-1} \frac{1 + i/N}{(1 + i/s)(1 + i/t)} , \quad i = 2, 3, \ldots .$$

The form holds for i = 2:

$$\alpha_2 = \alpha_1 \frac{N + 2}{(s+2)(t+2)} = \alpha_o k_1 \frac{N}{st} \frac{1 + 2/N}{(1+2/s)(1+2/t)}$$

$$= \alpha_o^2 k_1 \frac{1+2/N}{(1+2/s)(1+2/t)}$$

$\therefore k_2 = k_1 \frac{1+2/N}{(1+2/s)(1+2/t)}$, and the general form follows readily by induction.

C. Derivation of Equation (3) and (3a)

The exact expression for $V(n_o)$ has a structure similar to that for $E(N_o)$ containing the expressions α_i, $\frac{r_i}{r_1}$, (1-K), P_o and R'_m (remainder term somewhat different from R_m). Chapman's approximations lead to

$$V(N_o) \simeq N^2 \left[\frac{N}{st} + 7 \left(\frac{N}{st}\right)^2 + 38 \left(\frac{N}{st}\right)^3\right]$$

In accordance with the preceding discussion, we permit all but one approximation to stand to arrive at Eq. (3) and, with the same transformation as before, at Equation (3a).

D. $V(N_1)$

An exact form for $V(N_1)$, derived using Chapman's method, is

$$V(N_1) = \frac{(s+1)(t+1)(N+1)(N+2)}{(s+2)(t+2)}\,(1-K)\left[\frac{r_2}{r_1} + \frac{r_3}{r_1}\frac{N+3}{(s+3)(t+3)} + 2\frac{r_4}{r_1}\frac{(N+4)(N+3)}{(s+4)(t+4)(s+3)(t+3)} + \cdots\right.$$

$$\left. + (m-2)!\,\frac{r_m}{r_1}\frac{(N+m)\ldots(N+3)}{(s+m)\ldots(t+3)}\right] + (s+1)^2(t+1)^2(1-K)E(R_m'') - (N+1)^2(1-2K)$$

The same approximations as before reduce this to

$$V(N_1) \simeq \frac{(s+1)(t+1)(N+1)(N+2)}{(s+2)(t+2)}\left[1 + \frac{N+3}{(s+3)(t+3)} + 2\,\frac{(N+4)(N+3)}{(s+4)(t+4)(s+3)(t+3)} + \cdots\right.$$

$$\left. + (m-2)!\,\frac{(N+m)\ldots(N+3)}{(s+m)\ldots(t+3)}\right] - (N+1)^2$$

$$V(N_1) \simeq (s+1)^2(t+1)^2\left[\alpha_2 + \alpha_3 + 2\alpha_4 + \ldots + (m-2)!\,\alpha_m\right] - (N+1)^2 \qquad (5)$$

which can also be written as

$$V(N_1) \simeq (s+1)^2(t+1)^2\left(\frac{N}{st}\right)^2\left[(k_2 - k_1)2 + k_3\frac{N}{st} + 2k_4\left(\frac{N}{st}\right)^2 + \ldots + (m-2)!\,k_m\left(\frac{N}{st}\right)^{m-2}\right] \qquad (5a)$$

Appendix 2. Taylor's Series Derivation for $E(\hat{N})$ and $V(\hat{N})$

A. Let $\hat{N}$ be any estimate of N. $\hat{N}$ is a function of c, say w(c). Let $v(c) = [w(c) - N]^2$. Then $E[w(c)] = \sum_c w(c)P(c)$ and $E[v(c)] = \sum_c v(c)P(c)$ are respectively the mean and the mean-squared error of estimate $\hat{N}$. Consequently any method for evaluating the expected value of a function of a random variable can be used to find both $E(\hat{N})$ and $V(\hat{N})$.

B. Let the mean of c be m, its variance be σ^2 and its k^{th} central moment, $E[(c-m)^k]$, be μ_k. Let g(c) be a function (we will later let it be both w(c) and v(c)) which can be expanded in a Taylor's series about m:

$$g(c) = g(m) + (c-m)\, g'(m) + \frac{(c-m)^2}{2!} g''(m) + \frac{(c-m)^3}{3!} g'''(m) + \ldots$$

Multiply each term by P(c) and sum over all c. The result is

$$E[g(c)] = g + g'E(c-m) + \frac{g''}{2!} E[(c-m)^2] + \frac{g'''}{3!} E[(c-m)^3] + \ldots$$

where g and its derivatives are evaluated at $c = m$,

$$E(c-m) = 0$$

$$E[(c-m)^2] = \sigma^2$$

$$E[(c-m)^3] = \mu_3 .$$

Then

$$E[g(c)] \simeq g + \frac{\sigma^2}{2} g'' + \frac{\mu_3}{6} g''' + \frac{\mu_4}{4!} g^{(4)} \qquad (11)$$

If a truncated portion is to be a reasonable approximation of $E[g(c)]$, the series must converge, and rapidly. If g(c) does not vary too much near m, the derivatives will be small. But the $(c-m)^k P(c)$ terms must not be

too large; this requires that the domain of c not spread too far from m and/or that the remote points have very low probability.

C. We find $E(\hat{N})$ by replacing g(c) by $\hat{N} = w(c)$

$$E(\hat{N}) \simeq w + \frac{\sigma^2}{2} w'' + \frac{\mu_3}{6} w''' + \frac{\mu_4}{4!} w^{(4)} \qquad (12)$$

where w, w'' and $w^{(4)}$ are evaluated at c = m.

We find $V(\hat{N})$ by replacing g(c) and its derivatives by v(c) and its derivatives. Evaluating at m gives

$$v = (w-N)^2$$

$$v' = 2(w-N)w'$$

$$v'' = 2(w-N)w'' + 2(w')^2$$

$$v''' = 2(w-N)w''' + 6w'w''$$

$$v^{(4)} = 2(w-N)w^{(4)} + 8w'w''' + 6(w'')^2$$

Substituting these in Eq. (11) we find

$$V(\hat{N}) \simeq (w-N)^2 + \sigma^2 [(w')^2 + w''(w-N)] + \frac{\mu_3}{3} [3w'w'' + w'''(w-N)]$$

$$+ \frac{\mu_4}{12} [4w'w''' + 3(w'')^2 + w^{(4)}(w-N)]. \qquad (13)$$

D. Application to Estimate N_o

The mean and variance of c are known. The higher moments are not readily available but μ_3 can be calculated from the characteristic function of the hypergeometric distribution or from the formula for skewness [2]. μ_3 was in fact derived by the writer and substituted in Eqs. (12) and (13) to write expressions for $E(N_o)$ and $V(N_o)$. However the results in specific

cases were uniformly low indicating the need for more terms. We can avoid deriving μ_4* by using a normal approximation for $c[1]$ for which higher moments are more accessible.

In that event

$$\left.\begin{aligned} m &= \frac{st}{N} \\ \sigma^2 &= \frac{st}{N} \cdot \frac{(N-s)(N-t)}{N^2} = \frac{st}{N} q \\ \mu_3 &= 0 \\ \mu_4 &= 3\sigma^4 = 3\left(\frac{st}{N}\right)^2 q^2 \end{aligned}\right\} \qquad (14)$$

where $q = \frac{(N-s)(N-t)}{N^2}$

For N_o

$w(c) = st/c$	$w \equiv w(m) = N$
$w'(c) = -st/c^2$	$w' = -N(N/st)$
$w''(c) = 2st/c^3$	$w'' = 2N(N/st)^2$
$w'''(c) = -6st/c^4$	$w''' = -6N(N/st)^3$
$w^{(4)}(c) = 4!st/c$	$w^{(4)} = 4!N(N/st)^4$

Substituting in Eqs. (12) and (13), we obtain, finally,

$$E(N_o) \simeq N\left[1+q\left(\frac{N}{st}\right) + 3q^2\left(\frac{N}{st}\right)^2\right] \qquad (2)$$

$$V(N_o) \simeq N^2\left[q\left(\frac{N}{st}\right) + 9q^2\left(\frac{N}{st}\right)^2\right] \qquad (4)$$

---

* μ_3 was found to be $\frac{st}{N} \cdot \frac{(N-s)(N-t)}{N(N-1)} \cdot \frac{(N-2s)(N-2t)}{N(N-2)}$ but μ_4 does not follow the pattern of σ^2, μ_3; it is probably the sum of such a term and another.

E. Application to Estimate $\bar{\bar{N}}_o$

The form of $\bar{\bar{N}}_o = \frac{st}{\bar{c}}$ where $\bar{c} = \frac{1}{n} \sum_{i=1}^{n} c_i$ is identical with that of N_o. The only difference is that the random variable is $\bar{c}$ rather than c; the quantities m, σ^2, μ_3, μ_4 in Eqs. (12) and (13) must therefore be mean and central moments of $\bar{c}$. As the average of n random variables with identical distributions:

$$E(\bar{c}) = E(c) = \frac{st}{N}$$

$$\sigma^2(\bar{c}) = \frac{1}{n} \sigma^2(c)$$

Under the normal assumption for c, $\bar{c}$ is itself normal, and

$$\sigma^2(\bar{c}) = \frac{1}{n} \frac{st}{N} q$$

$$\mu_3(\bar{c}) = 0$$

$$\mu_4(\bar{c}) = 3\sigma^4(\bar{c}) = \frac{3}{n^2} \left(\frac{st}{N}\right)^2 q^2$$

We need only replace q by $\frac{q}{n}$ in Eqs. (2) and (4) to get the corresponding expressions for $\bar{\bar{N}}_o$:

$$E(\bar{\bar{N}}_o) = N\left[1 + \frac{q}{n} \left(\frac{N}{st}\right) + 3 \frac{q^2}{n^2} \left(\frac{N}{st}\right)^2\right] \tag{6}$$

$$V(\bar{\bar{N}}_o) = N^2\left[\frac{q}{n} \left(\frac{N}{st}\right) + 9 \frac{q^2}{n^2} \left(\frac{N}{st}\right)^2\right] . \tag{7}$$

Appendix 3. Confidence Intervals

A. Confidence Limits for N_o and N_1

Assume that c has a normal rather than hypergeometric distribution. The mean of the normal approximation [1] remains $\frac{st}{N}$ and the variance is only slightly changed: $\sigma^2 = \frac{st}{N} \cdot \frac{(N-s)(N-t)}{N^2}$. For a 100ϵ % confidence level, let $\lambda\sigma$ be the half-width of a symmetrical interval about the mean containing $(1-\epsilon)100\%$ of all occurrences. Then

$$P\left\{\frac{st}{N} - \lambda\sigma \leq c \leq \frac{st}{N} + \lambda\sigma\right\} = 1 - \epsilon \tag{8}$$

Known quantities are s, t, λ, ϵ and the experimentally determined c. The only unknown, when σ is replaced by the square root of σ^2 as given above, is N. From the left-hand inequality, we get

$$\frac{st}{N} - \lambda\sqrt{\frac{st}{N} \cdot \frac{(N-s)(N-t)}{N^2}} \leq c \ .$$

$$\frac{st}{N}\frac{(N-s)(N-t)}{N^2} \geq \frac{1}{\lambda^2}\left[\left(\frac{st}{N}\right)^2 + c^2 - 2c\left(\frac{st}{N}\right)\right]$$

$$st(N-s)(N-t) \geq \frac{1}{\lambda^2}\left[(st)^2 N + c^2 N^3 - 2cstN^2\right]$$

$$\frac{c^2}{\lambda^2}N^3 - N^2\left(\frac{2cst}{\lambda^2} + st\right) + N\left[\left(\frac{st}{\lambda}\right)^2 + s^2t + st^2\right] - (st)^2 \leq 0$$

$$g(N) \equiv N^3 - N^2\frac{st}{c}\left(2 + \frac{\lambda^2}{c}\right) + N\frac{st}{c^2}\left[st + \lambda^2(s+t)\right] - \left(\frac{st\lambda}{c}\right)^2 \leq 0 \tag{9}$$

From the right-hand inequality, we have

$$\frac{st}{N} + \lambda \sqrt{\frac{st}{N} \frac{(N-s)(N-t)}{N^2}} \geq c$$

$$\frac{st}{N} \frac{(N-s)(N-t)}{N^2} \geq \frac{1}{\lambda^2} \left[\left(\frac{st}{N}\right)^2 + c^2 - 2c\left(\frac{st}{N}\right) \right]$$

which is identical with the second expression above and therefore leads to the same result, namely $g(N) \leq 0$ where $g(N)$ is the polynomial in Equation (9).

Since $g(N) \leq 0$ represents both inequalities in (8), it is satisfied by all values of N in the confidence interval. The lower limit N_a of the confidence interval is characterized by the fact that smaller values of N are not in the interval and therefore do not satisfy $g(N) \leq 0$ but larger values are and do. Therefore N_a is the next integer at or below a solution of $g(N) = 0$ such that $g(N_a - 1) > 0$ and $g(N_a + 1) < 0$, i.e. near N_a, $g(N)$ changes from positive to negative with increasing N. Similarly the upper limit N_b of the confidence interval is the integer at or just above a larger root of $g(N) = 0$ where $g(N)$ changes from negative to positive with increasing N. In other words, the confidence limits are approximately two roots of $g(N) = 0$ between which $g(N)$ is negative. Inspection of $g(N)$ tells us that $g(0)$ is negative and that its derivative at 0 is positive. From Descartes' rule of signs, we know that $g(N) = 0$ has no negative and either one or three positive roots. From the genesis of the equation we know it has two positive roots since the interval limits do exist and are distinct. Therefore it has three real roots, all positive. It is apparent that $g(N)$ has the configuration shown in Fig. 5 and that the confidence limits are the two upper roots.

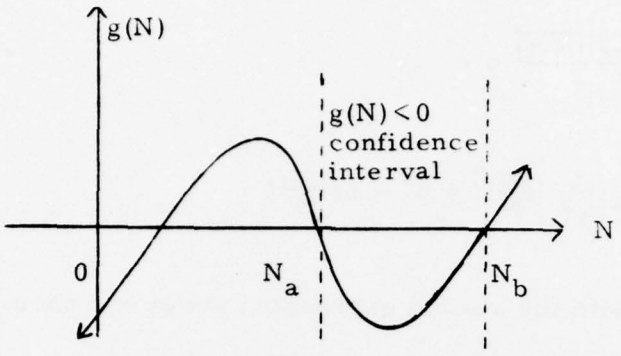


Figure 5. General Configuration of g(N)

B. Confidence Limits for $\overline{\overline{N}}_o$ and $\overline{\overline{N}}_1$

We begin with a set of experimental values $\{c_i : i = 1, \ldots, n\}$. Estimates $\overline{\overline{N}}_o$ and $\overline{\overline{N}}_1$ depend on the random variable $\bar{c} = \frac{1}{n} \sum_{i=1}^{n} c_i$, which is asymptotically normal with mean = $\frac{st}{N}$ and variance $\bar{\sigma}^2 = \frac{st}{nN} \frac{(N-s)(N-t)}{N(N-1)}$. Or, if we use the normal approximation for each c_i, $\bar{\sigma}^2 = \frac{st}{nN} \frac{(N-s)(N-t)}{N^2}$. Equation (8) becomes

$$P\left\{\frac{st}{N} - \lambda\bar{\sigma} \leq \bar{c} \leq \frac{st}{N} + \lambda\bar{\sigma}\right\} = 1 - \epsilon$$

Using the first form for $\bar{\sigma}^2$, the left-hand inequality leads to:

$$\frac{st}{N} - \lambda\sqrt{\frac{st}{nN}\,\frac{(N-s)(N-t)}{N(N-1)}} \leq \bar{c}$$

$$\lambda^2\,\frac{st}{nN}\,\frac{(N-s)(N-t)}{N(N-1)} \geq \left(\frac{st}{N}\right)^2 + (\bar{c})^2 - 2\bar{c}\,\frac{st}{N}$$

$$\lambda^2 st(N^2 - sN - tN + st) \geq (st)^2 n(N-1) + nN^2(N-1)\bar{c}^2 - 2stnN(N-1)\bar{c}$$

$$N^3 n\bar{c}^2 - N^2(n\bar{c}^2 + 2\,st n\bar{c} + \lambda^2 st) + N\left[(st)^2 n + 2st n\bar{c} + \lambda^2\, st(s+t)\right]$$

$$- (st)^2\,(n+\lambda^2) \leq 0$$

$$g(N) \equiv N^3 - N^2\left[1 + \frac{st}{\bar{c}}\left(2 + \frac{\lambda^2}{n\bar{c}}\right)\right] + N\,\frac{st}{\bar{c}}\left[2 + \frac{st}{\bar{c}} + \frac{\lambda^2(s+t)}{n\bar{c}}\right] - \left(\frac{st}{\bar{c}}\right)^2\left(1 + \frac{\lambda^2}{n}\right) \leq 0 \qquad (10)$$

The right-hand inequality leads to the same form. The reasoning described in part A of this appendix therefore establishes the confidence limits as the two largest roots of $g(N) = 0$ where $g(N)$ is as defined in Equation (10).

Appendix 4. Miscellaneous Proofs

A. Show that $\frac{1 + i/N}{(1 + i/s)(1 + i/t)} \leq \frac{1 + i/N}{(1 + i/\sqrt{st})^2}$

$(a-b)^2 = a^2 - 2ab + b^2 \geq 0$

$a^2 + b^2 \geq 2ab$

Let $a^2 = c$, $b^2 = d$; then $ab = \sqrt{cd}$

$c + d \geq 2\sqrt{cd}$

Add $1 + cd$ to both sides

$1 + c + d + cd \geq 1 + 2\sqrt{cd} + cd$

$(1 + c)(1 + d) \geq (1 + \sqrt{cd})^2$

Let $c = i/s$, $d = i/t$

Then $(1 + i/s)(1 + i/t) \geq (1 + i/\sqrt{st})^2$

And $\frac{1 + i/N}{(1 + i/s)(1 + i/t)} \leq \frac{1 + i/N}{(1 + i/\sqrt{st})^2}$

which was to be proved.

B. Show that $V = \text{var} + (\text{bias})^2$

Let $\hat{N}$ be any estimate of quantity N. Let m and b be the mean and bias respectively of $\hat{N}$: $m - N = b$ and $E(\hat{N} - m) = 0$

$$V(\hat{N}) = E[(\hat{N} - N)^2] = E\{[(\hat{N} - m) + (m - N)]^2\}$$

$$= E[(\hat{N} - m)^2] + (m - N)^2 + 2(m - N)\, E(\hat{N} - m)$$

$$= \text{var}(\hat{N}) + b^2$$

References

1. W. Feller, An Introduction to Probability Theory and Its Applications, Vol. 1, 2nd ed., John Wiley & Sons, Inc., New York, N. Y., 1957.
2. Handbook of Mathematical Functions, Ed. M. Abramowitz and I. A. Stegun, U. S. Dept. of Commerce, NBS Applied Mathematics Series, No. 55, 1964, p. 929.
3. D. G. Chapman, Some Properties of the Hypergeometric Distribution with Applications to Zoological Sample Census, Univ. of California Publications in Statistics, vol. 1 (1951), 131-160.
4. M. Lipow, Estimation of Software Package Residual Errors, TRW Systems Group, One Space Park, Redondo Beach, Calif. 90278, TRW-SS-72-09, Nov. 1972.
5. A. Papoulis, "Probability, Random Variables, and Stochastic Processes," McGraw-Hill, New York, 1965.
6. H. Cramér, Mathematical Methods of Statistics, Princeton Univ. Press, Princeton, 1946.

METRIC SYSTEM

BASE UNITS:

Quantity	Unit	SI Symbol	Formula
length	metre	m	...
mass	kilogram	kg	...
time	second	s	...
electric current	ampere	A	...
thermodynamic temperature	kelvin	K	...
amount of substance	mole	mol	...
luminous intensity	candela	cd	...

SUPPLEMENTARY UNITS:

plane angle	radian	rad	...
solid angle	steradian	sr	...

DERIVED UNITS:

Acceleration	metre per second squared	...	m/s
activity (of a radioactive source)	disintegration per second	...	(disintegration)/s
angular acceleration	*radian per second squared*	...	rad/s
angular velocity	radian per second	...	rad/s
area	square metre	...	m
density	kilogram per cubic metre	...	kg/m
electric capacitance	farad	F	A·s/V
electrical conductance	siemens	S	A/V
electric field strength	volt per metre	...	V/m
electric inductance	henry	H	V·s/A
electric potential difference	volt	V	W/A
electric resistance	ohm		V/A
electromotive force	*volt*	V	W/A
energy	joule	J	N·m
entropy	joule per kelvin	...	J/K
force	newton	N	kg·m/s
frequency	hertz	Hz	(cycle)/s
illuminance	lux	lx	lm/m
luminance	candela per square metre	...	cd/m
luminous flux	lumen	lm	cd·sr
magnetic field strength	ampere per metre	...	A/m
magnetic flux	weber	Wb	V·s
magnetic flux density	*tesla*	T	Wb/m
magnetomotive force	ampere	A	...
power	watt	W	J/s
pressure	pascal	Pa	N/m
quantity of electricity	coulomb	C	A·s
quantity of heat	joule	J	N·m
radiant intensity	watt per steradian	...	W/sr
specific heat	joule per kilogram-kelvin	...	J/kg·K
stress	pascal	Pa	N/m
thermal conductivity	watt per metre-kelvin	...	W/m·K
velocity	*metre per second*	...	m/s
viscosity, dynamic	pascal-second	...	Pa·s
viscosity, kinematic	square metre per second	...	m/s
voltage	volt	V	W/A
volume	cubic metre	...	m
wavenumber	reciprocal metre	...	(wave)/m
work	joule	J	N·m

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
1 000 000 000 000 = 10^{12}	tera	T
1 000 000 000 = 10^{9}	giga	G
1 000 000 = 10^{6}	mega	M
1 000 = 10^{3}	kilo	k
100 = 10^{2}	hecto*	h
10 = 10^{1}	deka*	da
0.1 = 10^{-1}	deci*	d
0.01 = 10^{-2}	centi*	c
0.001 = 10^{-3}	milli	m
0.000 001 = 10^{-6}	micro	μ
0.000 000 001 = 10^{-9}	nano	n
0.000 000 000 001 = 10^{-12}	pico	p
0.000 000 000 000 001 = 10^{-15}	femto	f
0.000 000 000 000 000 001 = 10^{-18}	atto	a

* To be avoided where possible.

*U.S. GOVERNMENT PRINTING OFFICE: 1977-714-025/115

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.