UNITED STATES AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

A FORTRAN PROGRAM FOR AN EXACT TEST OF INDEPENDENCE IN AN M X N CONTINGENCY TABLE

THESIS

GSA/MA/72-6 Marvin F. Schwartz, Jr.
Major USAF

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Air Force Institute of Technology (AFIT/EN)
Wright-Patterson AFB, Ohio 45433

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

A Fortran Program for an Exact Test of Independence in an M x N Contingency Table

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

AFIT Thesis

5. AUTHOR(S) (First name, middle initial, last name)

Marvin F. Schwartz, Jr.
Major USAF

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1972	55	33

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. N/A	GSA/MA/72-6
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release, distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
Approved for public release; IAW AFR 190-17 Keith A. Williams KEITH A. WILLIAMS, 1st Lt, USAF Assistant Director of Information, AFIT	

13. ABSTRACT

General m x n contingency tables are discussed along with relationships common to all such tables. The background for various probability models is presented as well as the theory for a general conditional test. The general test is then applied specifically to the probability model selected followed by a numerical example.

A brief history of past efforts in computing the exact test is provided. Finally, the major portions of the FORTRAN program are presented along with some illustrative examples. The final program is included as a useful entity.

DD FORM 1 NOV 68 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Fortran Program Exact Test of Independence Contingency Table						

GSA/MA/72-6

A FORTRAN PROGRAM FOR AN EXACT TEST OF INDEPENDENCE IN AN M X N CONTINGENCY TABLE

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

Marvin F. Schwartz, Jr.
Major USAF

Graduate Systems Analysis

March 1972

Preface

The procedure for an exact test of independence in general m x n contingency tables has been available for years. However, as with much of our knowledge, the application of theory in a practical sense has had to wait for the developement of high speed computers. The program presented here should be most useful in those instances where a test by approximation (although relatively simple and easily calculated) is either questionable or clearly inadvisable.

I wish to thank Dr. David R. Barr, my thesis advisor, for suggesting the topic and for his infinite patience in prodding me to the final solution.

Marvin F. Schwartz, Jr.

Contents

Contents

List of Figures

List of Tables

Abstract

General m x n contingency tables are discussed along with relationships common to all such tables. The background for various probability models is presented as well as the theory for a general conditional test. The general test is then applied specifically to the probability model selected followed by a numerical example.

A brief history of past efforts in computing the exact test is provided. Finally, the major portions of the FORTRAN program are presented along with some illustrative examples. The final program is included as a useful entity.

GSA/MA/72-6

A FORTRAN PROGRAM FOR AN EXACT TEST OF INDEPENDENCE IN AN M X N CONTINGENCY TABLE

I. Introduction

An Example

The May, 1970 issue of Aerospace Medicine contains an article which reports the following data on pilots who have made multiple ejections:

Table I
USAF Experience With Multiple Ejections

	Results of 1st Ejection	Results of 2nd Ejection		
Group	Number	Non-injured	Injured	Fatal
Non-injured	84	43	27	14
Injured	35	22	12	1
Totals	119	65	39	15

(From Ref 28)

This data represents the total United States Air Force experience with multiple ejections and as arranged forms the well known contingency table. Reference will be made back to this data later to show the analysis by approximation as reported and then the analysis by an exact test.

General M X N Contingency Tables

Suppose that K individuals are classified according to two criteria X and Y. Suppose further that there are m classifications

for X and n classifications for Y. A single individual would then be classified according to X_i ($i = 1, 2, \ldots, m$) and Y_j ($j = 1, 2, \ldots, n$). If the number of individuals belonging to both X_i and Y_j is A_{ij} we have an m x n contingency table. Such a table is shown in Figure 1.

	Y_1	Y_2	...	...	Y_n	Totals
X_1	A_{11}	A_{12}	...	...	A_{1n}	R_1
X_2	A_{21}	A_{22}	...	...	A_{2n}	R_2
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
X_m	A_{m1}	A_{m2}	...	...	A_{mn}	R_m
Totals	C_1	C_2	...	...	C_n	K

Figure 1. General m x n Contingency Table

Some Relationships

There are several relationships common to all contingency tables and essential to a discussion of the subject. If row totals are designated as R_i ($i = 1, 2, \ldots, m$) then $R_i = \sum_{j=1}^{n} A_{ij}$. Similarly, if column totals are designated as C_j ($j = 1, 2, \ldots, n$) then $C_j = \sum_{i=1}^{m} A_{ij}$. Further, $\sum_{i=1}^{m} R_i = K$, $\sum_{j=1}^{n} C_j = K$, and $\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij} = K$, where K is the total sample size. Let p_{ij} be the probability that an event falls in the cell corresponding to the ith row and jth column. Let p_i be the probability that an event falls into the ith row and q_j be the probability that an event falls

into the jth column. Then $p_i = \sum_{j=1}^{n} p_{ij}$, $q_j = \sum_{i=1}^{m} p_{ij}$ and $\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij} = 1$. Also, $\sum_{i=1}^{m} p_i = 1$ and $\sum_{j=1}^{n} q_j = 1$.

Extension to Higher Dimensions

Although the material presented here deals with just two criteria the only limit to the number of criteria which can be dealt with is one of practicality. If, for example, the elements of a population were to be classified according to three criteria X, Y, and Z with classifications X_i ($i = 1, 2, \ldots, r$), Y_j ($j = 1, 2, \ldots, s$), and Z_k ($k = 1, 2, \ldots, t$), a sample of K individuals could be classified in a three-way r x s x t contingency table. Thus there is no theoretical limit to the number of possible criteria and the treatment of the theory with added dimensions poses no special problems.

II. Testing For Independence

The Hypothesis of Independence

The contingency table is merely a convenient way to arrange data. The data itself can lead to numerous questions of cause and effect, or interaction (Ref 23:260). The question to be addressed here is one of no interaction, or independence. That is, is classification by one criterion independent of classification by the other criterion? Referring back to the general m x n table of Figure 1 we can ask if classification in any column is independent of the row entry of an individual. If the entries are independent then the probability of an individual being in a particular cell is merely the product of the probabilities of the row and column associated with that cell. The null hypothesis can therefore be formulated as $H_o: p_{ij} = p_i q_j$ $(i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$.

The probability model which will be used here is that which G. A. Barnard named as ' ; Independence Trial (Ref 2) which treats the table under consideration as though the marginal totals (i.e. row and column totals) are fixed in advance. Other models are the Double Dichotomy which treats the marginal totals as random variables and the Contingency Trial (or Homogeneity) which considers one set of marginals as fixed and the other set as random variables (Ref 2, 3, 4, 17:549-554; 27, 29, 33). The Independence Trial model is selected here since it is the only model which pro-

vides for an exact test and, from among the three models available, it provides the most powerful test (Ref 17:554; 29). Thus, by fixing the marginal totals, we will be using a conditional test based on the probability of a given table having fixed marginal totals. As will be shown, this conditional distribution is known and permits proceeding with an exact test.

In the vernacular of statistical hypothesis testing what is desired is a "two-tailed" test. Had the hypothesis been one which asked if entry in a particular cell was more or less likely to occur as a result of the associated row and column we would then have sought a "one-tailed" test. Here we are trying to determine (within limits) whether classification by one criterion is unaffected in either direction (high or low) by the other criteria (Ref 23:172). Hence, we require a "two-tailed" or non-directional test.

The Background for a Conditional Test

To construct a conditional test, consider the general situation of a random vector $\underset{\sim}{X} = (X_1, X_2, \ldots, X_n)$ with a distribution $f(\underset{\sim}{X};\underset{\sim}{\theta})$ and involving the unknown parameters $\underset{\sim}{\theta} = (\theta_1, \theta_2, \ldots, \theta_s)$. If $\underset{\sim}{\theta}$ has a set of joint sufficient statistics $\hat{\underset{\sim}{\Theta}}$, then the joint density of $\underset{\sim}{X}$ and $\hat{\underset{\sim}{\Theta}}$ may be written:

$$f(\underset{\sim}{x}, \hat{\underset{\sim}{\theta}}; \underset{\sim}{\theta}) = f_1(\underset{\sim}{x}|\hat{\underset{\sim}{\theta}}) f_2(\hat{\underset{\sim}{\theta}}; \underset{\sim}{\theta}) \qquad (1)$$

Now the conditional distribution of $\underset{\sim}{X}$ given $\hat{\underset{\sim}{\Theta}} = \hat{\underset{\sim}{\theta}}$ does not depend

on $\underset{\sim}{e}$ because the $\hat{\underset{\sim}{\Theta}}$ are joint sufficient statistics and for each value of $\hat{\underset{\sim}{e}}$ we can find numbers $A(\hat{e}_i)$ and $B(\hat{e}_i)$, $(i = 1, 2, \ldots, s)$ which are functions of $\hat{e}_i$, so that

$$\int_{A(\hat{e}_1)}^{B(\hat{e}_1)} \cdots \int_{A(\hat{e}_s)}^{B(\hat{e}_s)} f_1(\underset{\sim}{x}|\hat{\underset{\sim}{e}})d\underset{\sim}{x} = 1 - \alpha \qquad (2)$$

where α is some desired significance level for a critical region C. Then with some sample space S having an acceptance region A we have

$$P\left[A(\hat{\underset{\sim}{\Theta}}) < \underset{\sim}{X} < B(\hat{\underset{\sim}{\Theta}})\right] = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \int_{A(\hat{e}_1)}^{B(\hat{e}_1)} \cdots \int_{A(\hat{e}_s)}^{B(\hat{e}_s)} f_1(\underset{\sim}{x}|\hat{\underset{\sim}{e}}) f_2(\hat{\underset{\sim}{e}};\underset{\sim}{e}) d\underset{\sim}{x}\, d\hat{\underset{\sim}{e}} \qquad (3)$$

$$= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (1 - \alpha) f_2(\hat{\underset{\sim}{e}};\underset{\sim}{e}) d\hat{\underset{\sim}{e}}$$

$$= (1 - \alpha) \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_2(\hat{\underset{\sim}{e}};\underset{\sim}{e}) d\hat{\underset{\sim}{e}}$$

$$= 1 - \alpha$$

When X is a single variable and $\hat{\Theta}$ a single statistic the acceptance region A is as shown in Figure 2.

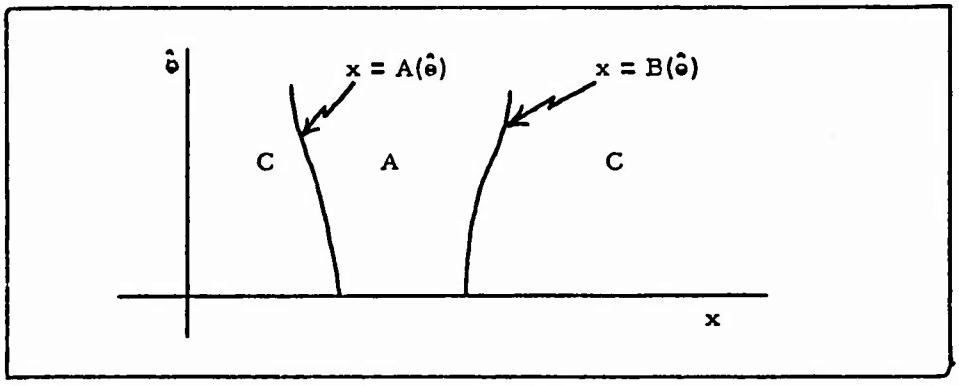


Figure 2. The Acceptance Region for a Conditional Test

Thus there exists, for each value of $\hat{e}$, an interval $(A(\hat{e}), B(\hat{e}))$ such that $P\left[A(\hat{e}) < X < B(\hat{e}) \mid \hat{\Theta} = \hat{e}\right] = 1 - \alpha$. Naturally, when this concept is extended to the multi-parameter case a geometric representation becomes impossible (Ref 25:258; 30:525).

Notice that this developement is dependent on formula (1) and requires the joint sufficient statistics $\underset{\sim}{\hat{\Theta}}$ in order to make the calculations shown. If $f_1(\underset{\sim}{x}|\underset{\sim}{\hat{e}})$ were not free of $\underset{\sim}{e}$, then calculation of the probability would depend on unknown parameters and the procedure used above would be inappropriate.

The Test as Employed

The hypothesis of independence was stated as, H_o: $p_{ij} = p_i q_j$. Unfortunately, the parameters p_i and q_j are unknown. However, it can be shown that the estimators R_i/K $(i = 1, 2, \ldots, m)$ and C_j/K $(j = 1, 2, \ldots, n)$ form a set of joint sufficient statistics for p_i and q_j respectively. By a one for one transformation, R_i $(i = 1, 2, \ldots, m)$ and C_j $(j = 1, 2, \ldots, n)$ are also sufficient statistics (Ref 25:171) and we can determine the exact probability for any contingency table by means of the following formula:

$$P_L = \frac{\prod_{m=1}^{k} \prod_{i_m=1}^{r_m} a_{m i_m}!}{(N!)^{k-1} \prod_{i_1=1}^{r_1} \prod_{i_2=1}^{r_2} \cdots \prod_{i_k=1}^{r_k} l_{i_1 i_2 \cdots i_k}!} \qquad (4)$$

where P_L is the conditional probability of an $(r_1 \times r_2 \times \cdots \times r_k)$

K-variate contingency table $L \equiv (l_{i_1 i_2 \cdots i_k})$, where $l_{i_1 i_2 \cdots i_k}$ is the entry in the cell identified by the K indices i_m given all the marginal totals

$$a_{m i_m} = \sum_{i_1=1}^{r_1} \cdots \sum_{i_{m-1}=1}^{r_{m-1}} \sum_{i_{m+1}=1}^{r_{m+1}} \cdots \sum_{i_k=1}^{r_k} l_{i_1 i_2 \cdots i_k} \qquad (5)$$

This formula was first presented by Freeman and Halton (Ref 10) and a more recent derivation is presented by Halton (Ref 13). Eq (4) appears in more restricted forms in the literature and for use here with just two criteria can be reduced to:

$$P_L = \frac{\prod_{i=1}^{m} \prod_{j=1}^{n} r_i! c_j!}{k! \prod_{i=1}^{m} \prod_{j=1}^{n} a_{ij}!} \qquad (6)$$

using the notation first presented. Numerous derivations are available for Eq (6). The derivation for a 2 x 2 table was first published by Fisher in 1934 (Ref 7). Irwin derived it independently in 1935 and the test as applied to a 2 x 2 table is often referred to as "The Fisher-Irwin Exact Test". Most derivations are based on a likelihood ratio test statistic (Ref 25:313-316). The right side of Eq (6) is the multivariate hypergeometric distribution and gives the conditional distribution of the A_{ij} given R_i and C_j. Since the last cell entry in each row can be determined from the first (n-1) entries in that row and the row total, and the last cell entry in each column is fixed by the first (m-1) entries and the column total, the distribution involves (m-1) x (n-1) independent variables. Eq (6) can

thus be substituted into a discrete form of Eq (2) to determine an appropriate acceptance region. In applying the exact test to a general m x n table Freeman and Halton (Ref 10) have suggested the following rule for computing an exact P value (Ref 12). The conditional probability of an observed table can be computed from Eq (6). The conditional probability of observing any other table having the same marginal totals is

$$P(\underset{\sim}{X}=\underset{\sim}{x}|\underset{\sim}{R}=\underset{\sim}{r},\ \underset{\sim}{C}=\underset{\sim}{c}) = \frac{\prod_{i=1}^{m}\prod_{j=1}^{n} r_i!c_j!}{k!\prod_{i=1}^{m}\prod_{j=1}^{n} x_{ij}!} \tag{7}$$

where $\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = k$, $\sum_{i=1}^{m} x_{ij} = c_j$, and $\sum_{j=1}^{n} x_{ij} = r_i$. We consider all possible configurations of such $\underset{\sim}{x}$'s such that

$$P(\underset{\sim}{X}=\underset{\sim}{x}|\underset{\sim}{R}=\underset{\sim}{r},\ \underset{\sim}{C}=\underset{\sim}{c}) \leq P(\underset{\sim}{A}=\underset{\sim}{a}|\underset{\sim}{R}=\underset{\sim}{r},\ \underset{\sim}{C}=\underset{\sim}{c}) \tag{8}$$

Then $P = \sum_{i \in I} P(\underset{\sim}{X}_i=\underset{\sim}{x}_i|\underset{\sim}{R}=\underset{\sim}{r},\ \underset{\sim}{C}=\underset{\sim}{c})$ where the sum is with respect to the $\underset{\sim}{x}$'s subject to the constraint (8).

Kendall and Stuart (Ref 17:552) give a numerical illustration using the following table:

Table II
Illustrative Example

4	16	20
1	21	22
5	37	42

(From Ref 17)

Given fixed marginal totals the probability that $A_{11} = 4$ is

$$\frac{c_1!c_2!r_1!r_2!}{k!a_{11}!a_{12}!a_{21}!a_{22}!} = \frac{5!37!20!22!}{42!4!16!1!21!} = .1253$$

Probabilities for all possible X_{11} given r_1, r_2, c_1, c_2 are plotted in Figure 3.

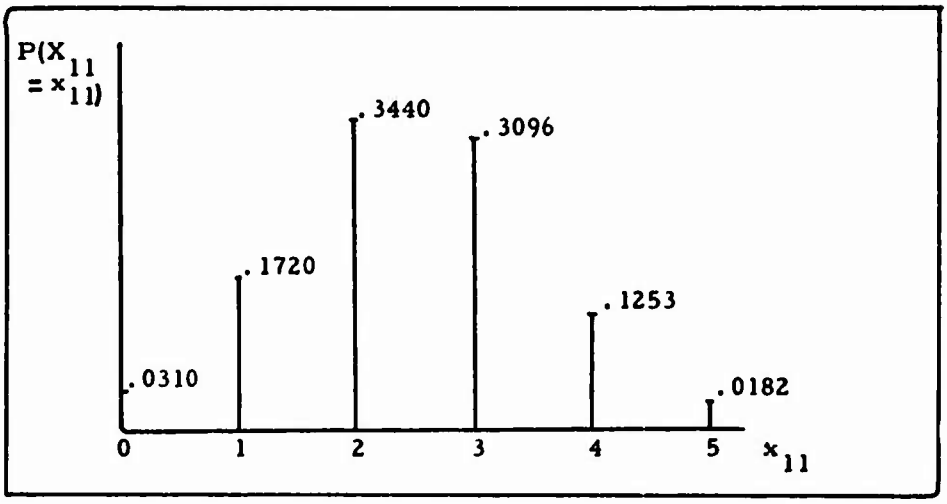


Figure 3. $P(X_{11} = x_{11})$ versus x_{11}

$P = \sum_{i \in I} P(X_{11_i} = x_{11_i} | \underset{\sim}{R}=\underset{\sim}{r}, \underset{\sim}{C}=\underset{\sim}{c})$ where the sum is over the x's subject to the constraint (8) is .1745. Thus at a significance level of .05 or .10, for example, we would fail to reject the hypothesis of independence for the given table. The distribution is a univariate hypergeometric and is unimodal while the test is non-directional. An observed table with the same marginal totals but with A_{11} equal to 0 or 5 would have led to rejection of the hypothesis at a significance level of .05.

III. The Need for a Computer Program

Chi-square and its Limitations

As can be seen, the formula for computing an individual probability can lead to an extremely laborious calculation. In words,

$$P_L = \frac{\text{Product of all (border totals!)}}{(K!) \times \text{product of all (cell totals!)}} \tag{9}$$

In all but the most trivial problems the time required to perform the calculations without resorting to computers could quickly become prohibitive. The common recourse is to fall back on an approximation of one form or another. For example, it can be shown that for large samples if the distribution (6) is replaced by its multivariate normal approximation, the criterion

$$u = \sum_{ij} \frac{\left[A_{ij} - (R_i C_j / k)\right]^2}{R_i C_j / k} \tag{10}$$

approaches the chi-square distribution asympotically (Ref 25). Wilks (Ref 32) derives a similar statistic with an approximate chi-square distribution. However, the chi-square approximation is limited in its applicability and various rules of thumb prevail concerning its use with most based on the expected frequency for any A_{ij} as computed by

$$E = R_i C_j / k \tag{11}$$

Most commonly, the approximation is discouraged when the expected frequency in any cell is less than five (Ref 11:295). Kenney and Keeping (Ref 18:307) recommend the exact method when the smallest expected frequency in any cell is less than 10 while Mood (Ref 25:413) considers the approximation "fairly good" in a 2 x 2 table if both column totals exceed 10. Pierce (Ref 26:115) remarks that, "...the computational convenience of the chi-square model is no longer a plausible excuse for the 2 x 2 case..."

Other Limited Aids

Exact treatment of the 2 x 2 case has certainly received a great deal of attention. Best known, perhaps, is Lieberman and Owen's Tables of the Hypergeometric Probability Distribution (Ref 21). These were preceded by Mainland and Murray (Ref 24) and Finney et al (Ref 6) have provided a more extensive set. Still another set has been constructed by Clark (Ref 4). Leslie (Ref 19) has written up a procedure using binomial coefficients which is valid for both one and two-tailed tests.

Beyond the very limited case of a 2 x 2 table much less has been done. Leyton (Ref 20) treats the 2 x 3 table using Leslie's approach of binomial coefficients and the critical region suggested by Freeman and Halton (Ref 10). Pierce (Ref 26) discusses extension beyond the 2 x 2 case and has written an ALGOL and a BASIC language computer program for the 2 x 3 case. As regards the

BASIC language program the prime algorithm is correct and the point probability computed for the observed table is accurate. However, the cumulative probabilities are inaccurate and should not be relied on. The program finds and stores exact probabilities for various possible arrays. The stored probabilities are then scanned in the order in which they were computed until the first time a probability is found which exceeds the point probability of the observed table. The program is then immediately terminated with an output which is purported to be, "The probability of a distribution as probable or less probable than that observed." Since the program terminates abruptly and does not consider all of the probabilities in the conditional sample space some probabilities "...less probable than that observed..." are usually lost and not reported. No theory or explanation is given for doing this and yet the program is classed as "non-directional".

The program reported here looks at all probabilities according to Eq (6) and is applicable to any size m x n table limited only by computer capabilities.

A Questionable Finding

Consider again the data on multiple ejections cited earlier. For convenience the data is repeated below. The researcher reported that, "However, out of 84 individuals who had received no reportable injury on the first ejection, there were 14 fatalities on the

Table III
USAF Experience With Multiple Ejections

Group	Results of 1st Ejection Number	Results of 2nd Ejection		
		Non-injured	Injured	Fatal
Non-injured	84	43	27	14
Injured	35	22	12	1
Total	119	65	39	15

(From Ref 28)

second ejection. A chi-square test applied to this finding revealed it to be highly significant at the 95% level."

If one were to test for independence of classification for this table Eq (11) could be applied to obtain the expected frequency in the A_{23} cell. Here, $E = (35) \times (15)/119 = 4.4$ and a chi-square approximation is questionable.

IV. Developing the Program

Basic Considerations

In developing the program the basic considerations were:

1. The probability of the given array must be established and all other probabilities in the sample space compared with it.

2. Each feasible array, given the fixed marginal totals, must be determined.

3. The probabilities associated with each feasible array must be computed and compared with the probability of the given array.

Each of these considerations is discussed in detail and summarized by a numerical example to clarify the discussion.

The Probability of the Given Array

The computation for the probability of the given array should employ Eq (6). Since the result is a probability we know that $0 \leq P_L \leq 1$. Although the equation appears as a simple fraction, the number of factors in the numerator is always less than the number of factors in the denominator, each factor is a factorial, and, provision must be made for the fact that $0! = 1! = 1$. An obvious approach to the computation is to determine the individual values for both the numerator and denominator and find their quotient. This procedure could quickly overflow in many computers. Consider just the factor k! in the denominator of the expression. $20! = 2.43 \times 10^{18}$ and it is obvious that some extraordinarily large

values could appear long before the final division of numerator by denominator. Since the sum of all of the marginal totals (appearing in the numerator) equals the sum of each cell entry plus the sample size a scheme was sought which would move from numerator to denominator in such a way as to keep any intermediate result as small as possible. The scheme decided upon requires that the array be arranged as follows:

1. If the array is not square the number of rows must be less than the number of columns.

2. From top to bottom, each row total must be less than or equal to the next succeeding row total.

3. From left to right, each column total must be less than or equal to the next succeeding column total.

The algorithm takes the final form of:

$$P = P(\frac{a}{ak})(\frac{b}{d}) \tag{12}$$

starting with P = 1.

a represents the row totals. It is set at one, incremented by one up to the value of r_1, reset to one and incremented up to the value of r_2, etc.

b represents the column totals. It is set to one and increased by one after each n repetitions (where n = number of columns) until achieving a value of c_1. It is then increased after n-1 repeti-

tions with the number of repetitions between increments being reduced each time b takes on the value of the succeeding column. Thus the numerator is kept from increasing too rapidly so that intermediate values of P are, except in unusual instances, kept to a value less than one. In this way there is little likelihood of exceeding the upper limits on any computer. Never-the-less, intermediate values are periodically looked at and adjusted, if necessary, by the scaling procedure described below.

ak represents each cell value. It is set at one and is incremented by one up to the value of a_{11}, reset to one, incremented up to the value of a_{12}, etc. Any A_{ij} equal to zero is ignored since it does not affect any marginal totals or the sample size and its factorial value of one does not alter the calculations.

d represents the total sample size and is incremented by one for each successive calculation. The computation is complete when a has been incremented to the final value of r_m.

The final value of P must be less than one. However, it is this very point which creates the most formidable problem of all. As the number of cells and/or the sizes of the marginal totals increase, the number of points in the sample space increase astronomically while corresponding probabilities get unbelievably small. Consider the following examples:

Table IV
An Example

2	0	3	5
1	6	5	12
3	6	8	17

(From Ref 10)

a. The probability of the given array is .027149. The number of possible arrays, given the marginal totals, is 18.

Table V
An Example

2	0	1	3
1	2	2	5
2	3	4	9
5	5	7	17

b. The probability of the given array is .023139. Although the sample size remains the same, the number of possible arrays, given the marginal totals, is 179.

Table VI
An Example

11	74	181	22	288
1	25	201	109	336
12	99	382	131	624

(From Ref 5)

c. The probability of the given array is $.77316 \times 10^{-23}$. Among the 171,600 possible arrays is one with a probability of $.49981 \times 10^{-94}$.

Quite obviously the situation exists where many machines would have an exponent underflow. Additionally, in trying to add numbers such as these a sub-total could conceivably be reached to which each additional value would be too small to be significant and would thus be lost although the true total of thousands of these minute values would total up to a significant sum. The problem can be overcome by scaling each value and considering the individual probabilities as two-part numbers. Hence, PSTOR(1) could represent

a value between one and 10^{-10}, PSTOR(2) would represent values from 10^{-10} to 10^{-20}, and PSTOR(S) would be a value between $10^{-10(S-1)}$ and $10^{-(10 \times S)}$. The computer merely takes a value like 10^{-47}, multiplies it by 10^{40}, and stores it as PSTOR(5) = 10^{-7}. Additions to PSTOR(5) would be continued until the total exceeds one. The value is then multiplied by 10^{-10} and put in location PSTOR(4). At the completion of a run, the lowest ten locations are accumulated and PSTOR(1) is output as a final result. Any values remaining in other locations could only affect the result by a trivial amount and can be ignored. The program could easily be modified to change scaling factors so as to decrease roundoff error depending on the machine to be used.

One additional step is required when this first probability is computed. It must be used for comparison with all other probabilities computed later on. Some decision must be made as to the amount of accuracy desired, consistent with machine capabilities. The present program was tested on a CDC 6600 with 29 significant digits in double precision. By taking the initial probability, multiplying by 10^{10}, and truncating to an integer value, all subsequent probabilities can be compared to 10 decimal places of accuracy. Any probability with the same scaling factor as that originally computed is similarly manipulated for comparison. This procedure was tested with example b above, and had zero roundoff error. The test printed out all possible arrays with their associated prob-

abilities, and the original array, reencountered after 108 trials, showed the identical probability although several different algorithms had been used in the interim, and some roundoff error would have been reasonable.

Finding All Feasible Arrays

Within the constraints of fixed marginal totals each feasible array must be determined. Since the marginal totals are fixed the last cell in each row can be determined from the first n-1 cells in that row and the row total. Similarly, the last cell in each column can be determined from the first m-1 cells and the column total. Thus:

$$a_{in} = r_i - \sum_{j=1}^{n-1} a_{ij} \qquad \text{for all } i \tag{13}$$

and,

$$a_{mj} = c_j - \sum_{i=1}^{m-1} a_{ij} \qquad \text{for all } j \tag{14}$$

Hence, (n-1) x (m-1) cells and the marginal totals must be specified to completely determine any array. Obviously no cell value can be less than zero nor larger than the smaller total for the row or column associated with that cell. The bounds on any cell value are, therefore,

$$0 \leq a_{ij} \leq \min(r_i, c_j), \quad (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n) \tag{15}$$

Notice that the (mn) cell is somewhat unique. Its value can be

determined as:

$$a_{mn} = r_m - \sum_{j=1}^{n-1} a_{mj} \tag{16}$$

or as:

$$a_{mn} = c_n - \sum_{i=1}^{m-1} a_{in} \tag{17}$$

Thus, in the first m-1 rows values can be assigned to all cells but the last. The remaining value in each row is determined from Eq (13). The first n-1 values in the last row are then determined from Eq (14) and a_{mn} can be determined from Eq (16) or Eq (17). (The program uses Eq (17).). To simplify the discussion, the (m-1) x (n-1) cell values to be altered can be referred to as changeable values. These are selected from the first n-1 cells in each of the first m-1 rows. In moving from one changeable value to the "next" we will move from left to right and top to bottom so that the last changeable value in a row is $a_{i,\,n-1}$. The "next" changeable value will be $a_{i+1,\,1}$ and the last changeable value in the array is $a_{m-1,\,n-1}$.

Suppose the changeable values can take on a maximum value of $v_1, v_2, \cdots, v_k$ where $k = (m-1) \times (n-1)$. Then the maximum number of arrays will be $(v_1 + 1) \times (v_2 + 1) \times \cdots \times (v_k + 1)$. If, for example, k was equal to 3 we could set the last two changeable values to zero and vary the first from 0 to v_1. Then with the second value increased by one we could again vary the first value through its range of

0 to v_1. This would be repeated each time the second value was increased until a further increase would exceed v_2. At this time, v_3 could be set to one while v_1 and v_2 were set to zero and all of the preceeding steps would be repeated for each increment of the third value until it could no longer be increased without exceeding its limit of v_3. Thus, $(v_1 + 1) \times (v_2 + 1) \times (v_3 + 1)$ possible combinations would be generated. This is essentially the procedure used by the program except that each change in a changeable value must be followed by checks to determine if all of the a_{in}'s, the a_{mj}'s, and the a_{mn} are positive.

Initially, all of the changeable values are set to zero. By Eq (13) $a_{in} = r_i$ $(i = 1, 2, \ldots, m-1)$, by Eq (14) $a_{mj} = c_j$ $(j = 1, 2, \ldots, n-1)$, and a_{mn} is then computed by Eq (17). If a_{mn} is non-negative the array is feasible and its probability is computed. Whether or not a_{mn} is non-negative, the first changeable variable (a_{11}) is incremented by one, a_{1n} and a_{m1} are each decreased by one and a_{mn} is increased by one. Any time these last three values are non-negative the array is feasible. Once a_{1n} or a_{m1} becomes negative any further increase in a_{11} is futile and it is reset to zero. The next changeable value is now incremented by one and checks for non-negativity are conducted, and a_{11} is again cycled until a_{1n} or a_{m1} goes negative. The next changeable value is incremented again and the necessary checks are made. Each changeable value in turn is incremented until the last cell in the associated row or

column goes negative. The incremented value which caused this is then set to zero and the next changeable value is incremented by one. Notice that, except for a_{11}, a value is never incremented unless the preceeding value is set to zero and, also, whenever a value is incremented, all previous values must be at zero. The procedure is complete when the last changeable value cannot be incremented further due to the restriction of Eq (15).

The program takes advantage of one additional relationship to shorten processing time. Any time a single a_{ij} ($i = 1, 2, \ldots, m-1$; $j = 1, 2, \ldots, n-1$) is incremented the only other changes will be a decrease in the value of the associated a_{in} and a_{mj} and an increase in a_{mn}. Therefore, if a feasible array has been found and the next increment in a changeable value does not cause a negative value for the associated a_{in} or a_{mj} the new array is feasible and a_{mn} need not be checked since it must still be non-negative. (Conversely, a_{mn} can never decrease in value unless a changeable value is reset to zero.) The complete sequencing for the example from Table IV is demonstrated in Table VII on the following page.

The Probability of Each Succeeding Array

The first feasible array encountered has its probability computed using the same procedure developed for determining the probability of the given array. Each successive feasible array could use the same algorithm but this would be extremely time consuming

Table VII
Sequencing of the Example From Table IV

(1)	0	0	5	(2)	1	0	4	(3)	2	0	3	(4)	3	0	2
	3	6	3		2	6	4		1	6	5		0	6	6
(5)	0	1	4	(6)	1	1	3	(7)	2	1	2	(8)	3	1	1
	3	5	4		2	5	5		1	5	6		0	5	7
(9)	0	2	3	(10)	1	2	2	(11)	2	2	1	(12)	3	2	0
	3	4	5		2	4	6		1	4	7		0	4	8
				(13)	0	3	2	(14)	1	3	1	(15)	2	3	0
					3	3	6		2	3	7		1	3	8
				(16)	0	4	1	(17)	1	4	0	(18)	0	5	0
					3	2	7		2	2	8		3	1	8

(From Ref 10)

and much easier methods are available.

If an a_{ij} is incremented by one and the array is feasible then the only other changes in the array will be that a_{mn} is increased by one and a_{in} and a_{mj} are decreased by one. The probability P_{t+1} of this new array is simply the previous probability P_t multiplied by:

$$\frac{(a_{in}+1)(a_{mj}+1)}{(a_{ij})(a_{mn})} \tag{18}$$

where the a's are current values in the new array. If P_t is treated as though it had been computed by Eq (6) then Eq (18) reflects the change in going to P_{t+1} caused by the change in four cells in the denominator. a_{ij} and a_{mn} are increased while a_{in} and a_{mj} are

decreased. This decrease is taken care of in the numerator of (18).

If more than one of the a_{ij}'s changes a similar procedure is used to compute the probability of the next feasible array starting with the previously computed probability. Consider first, the effect of a single cell which can decrease in value. Let $a_{pq}^* = a_{pq} - d$, where: a_{pq}^* is the new cell value, a_{pq} is the old cell value and d is the difference in their values. Then $1/a_{pq}!$ would be equal to $1/(a_{pq} \times (a_{pq}-1) \times \cdots \times (a_{pq}-d+1) \times a_{pq}^*!)$ and $1/a_{pq}^*!$ would equal $\prod_{v=0}^{d-1}(a_{pq}-v) \times (1/a_{pq}!)$. Hence the previous probability must be multiplied by $\prod_{v=0}^{d-1}(a_{pq}-v)$ to account for the effect of the decrease in a_{pq}. If, for example, a probability had been computed with $a_{pq} = 15$ and in the next feasible array, a_{pq} were 12, then the change for this cell would change the probability by a factor of, 15x14x13. Had a_{pq} increased by d to a value a_{pq}^* then the old probability would have to be divided by $\prod_{v=1}^{d}(a_{pq}+v)$. Thus, if more than one of the a_{ij}'s $(i = 1, 2, \ldots, m-1; j = 1, 2, \ldots, n-1)$ changes the new and old values are stored as well as the new and old values for any changes to any a_{mj} or a_{in}. The probability for the previous array is then adjusted for each cell which has changed by using the procedure just discussed.

The probability associated with each feasible array is now compared with the probability of the given array. If the present scaling factor is smaller than the original one, the associated probability is greater than that of the given array. These values are never-the-

less accumulated for a final check to see that all probabilities have been determined by looking for an overall sum of one. For scaling factors equal to the factor associated with the given array, the digits are compared as previously described. Scaling factors greater than that associated with the given array indicate immediately a lesser probability.

A Partial Numerical Example

Consider example b, from before with the array from Table V

2	0	1	3
1	2	2	5
2	3	4	9
5	5	7	17

With four independent variables all cell values can be determined from the marginal totals and the values a_{11}, a_{12}, a_{21}, and a_{22}. The probability of the given array is .023139 and the scaling factor is 10^0. a_{11}, a_{12}, a_{21}, and a_{22} are set to zero and the array becomes:

0	0	3	3
0	0	5	5
5	5	-1	9
5	5	7	17

The negative entry is non-feasible and a_{11} is incremented by one giving:

1	0	2	3
0	0	5	5
4	5	0	9
5	5	7	17

The probability for this array is .000077129. This is less than .023139 and is stored in PSTOR(1).

a_{11} is increased by one to give:

2	0	1	3
0	0	5	5
3	5	1	9
5	5	7	17

The probability for this array is:

$$P = .000077129 \times (2/2) \times (4/1) = .00030851$$

This probability is again less than .023139 and is added in to the value in PSTOR(1). a_{11} is again increased by one to give:

3	0	0	3
0	0	5	5
2	5	2	9
5	5	7	17

The probability for this array is:

$$P = .00030851 \times (1/3) \times (3/2) = .00015426$$

After computing and comparing probabilities the next increase in a_{11} gives:

4	0	-1	3
0	0	5	5
1	5	3	9
5	5	7	17

The negative entry is non-feasible so a_{11} is reset to zero and a_{12} is incremented by one. The new array becomes:

0	1	2	3
0	0	5	5
5	4	0	9
5	5	7	17

Now a_{11} has decreased from 3 to 0, a_{32} has decreased from 5 to 4, and a_{33} from 2 to 0. a_{12} increased from 0 to 1, a_{13} from 0 to 2,

and a_{31} from 2 to 5. The new probability is:

P = .00015426 x (3x2x1/1) x (5/2x1) x (2x1/5x4x3) = .000077129

To illustrate other intermediate steps, sequencing for feasible arrays goes from:

0	3	0		0	0	3		1	0	2
0	0	5	to	1	0	4	to	1	0	4
2	2			4	5	0		3	5	1

and from:

0	3	0		0	0	3		1	0	2
4	1	0	to	0	2	3	to	0	2	3
1	1	7		5	3	1		4	3	2

The final computation is on the array,

3	0	0	3
0	5	0	5
2	0	7	9
5	5	7	17

The final outcome showed 179 feasible arrays. Computation was done in double precision with results printed out to 18 significant digits. On the above example, the probability of an array equal to or less than the given array was .776840806252570958. The probability of an array more probable than the given array was reported as .223159193747429042. Their sum was 1.000000000000000000.

Final Considerations

As alluded to previously, the number of feasible arrays, given fixed marginal totals, increases explosively as the dimensions and sample size of any given table increase. The number of possible

arrays can be found by:

$$\sum_{n_1, n_2, \ldots, n_k} n! \prod_{i=1}^{k} \frac{x_i^{n_i}}{n_i!} \tag{19}$$

where the x_i are cell entries in the range of 0 to n and $\sum_{i=1}^{k} n_i = n$ (Ref 25:26, 27). The summation in Eq (19) is then taken over all sets of values of $n_1, n_2, \ldots, n_k$ such that their sum is n and each n_i is an integer in the range of 0 to n inclusive. Finding this value is a formidable task in itself and its solution does not provide any real guidance in estimating computer run time. Some tables sequence quite readily from one feasible array to the next and run time is relatively short. Other tables may have fewer possibilities but require much more checking and rearranging with a resultant increase in run time. As a consequence, every effort was made to minimize run time while relying on the FORTRAN IV language (Ref 22) which is applicable to most machines. As a result, time saving steps available to FORTRAN EXTENDED (such as mixed mode arithmetic) were purposely avoided.

The program was checked on a CDC 6600 and use on a smaller machine may require more frequent checks for scaling than is now accomplished. Additionally, increased accuracy could be sought by resorting to double precision. The present program is in single precision since this provided a savings in time of approximately 40%. The CDC 6600 can operate in the range of 10^{-308} to 10^{337}.

Real constants have a maximum of 15 decimal digits while double precision constants may be a maximum of 29 decimal digits. Consequently, use of the program on a smaller machine may result in exponent underflows or loss of significance. The present program can be adapted to smaller machines with minor changes.

A Summary Example

Return again to the data on multiple ejections and the researcher's results following Table III. The computer program reported:

1. 456 possible arrays.

2. The probability of an array equal to or less than the given array is .104957718232142.

3. The probability of an array greater than the given array is .895042281766887.

4. As a check, the sum of all probabilities is .999999999999030.

Hence, a "two-tailed" test of independence at the .10 level would fail to reject and no statement is appropriate (at the .10 level) concerning a greater or lesser likelihood of fatality on a second ejection following non-injury on the first ejection.

V. The Resulting Program

Appendix A is a flow chart for the program. The program itself is included as Appendix B. Comments are included so that the program is an entity in itself and can be used independently of the report contained here.

Program run time can become quite large and is dependent upon the number of rows and columns as well as the sample size. For instance, example a in the text took approximately .2 seconds, while example b took .25 seconds. Example c, with a fairly large sample size, took approximately 12 seconds. A 3 x 5 array with all row totals equal to five and all column totals equal to three takes about one second.

A final note of caution is in order. The table must be arranged as specified in the comments. Failure to do so may return incorrect answers.

Bibliography

1. Abbott, W. J. "Reliability of Clinical Data - A Feasible Method for Performing Fisher's Exact Test." Texas Journal of Science, 18:157-164 (1966).

2. Barnard, G. A. "Significance Tests for 2 x 2 Tables." Biometrika, 35:123-137 (1947).

3. Bhapkar, V. P., and G. G. Koch. "On the Hypothesis of 'No Interaction' in Contingency Tables." Biometrics, 24:567-594 (1968).

4. Clark, R. E. Critical Values in 2 x 2 Tables. University Park, Pennsylvania: The Pennsylvania State University, August 1969.

5. Drotning, J. E., and D. B. Lipsky. "The Effectiveness of Reinstatement as a Public Policy Remedy: The Kohler Case." Industrial and Labor Relations Review, 22:179-198 (1969).

6. Finney, D. J., R. Latcha, B. M. Bennett, and C. Horst. Supplement to Tables for Testing Significance in a 2 x 2 Table. Cambridge, England: Cambridge University Press, 1966.

7. Fisher, R. A. Statistical Methods for Research Workers. Edinburgh: Oliver and Boyd, 1934.

8. -----. "The Interpretation of Experimental Four-Fold Tables." Science, 94:208-217 (1941).

9. Fraser, D. A. S. "Sufficient Statistics With Nuisance Parameters." The Annals of Mathematical Statistics, 27:838-842 (1956).

10. Freeman, G. H., and J. H. Halton. "Note on an Exact Treatment of Contingency, Goodness of Fit and Other Problems of Significance." Biometrika, 38:141-149 (1951).

11. Freund, J. E. Modern Elementary Statistics, 3rd Ed. New Jersey: Prentice-Hall, 1967.

12. Gart, J. J. "Alternative Analysis of Contingency Tables." Journal of the Royal Statistical Society, B, 28:164-183 (1966).

13. Halton, J. H. "A Rigorous Derivation of the Exact Contingency Formula. " Proceedings of the Cambridge Philosophical Society, 65:527-530 (1969).

14. Healy, M. J. R. "Exact Tests of Significance in Contingency Tables." Technometrics, 11:393-395 (1969).

15. Hodges, J. L., Jr., and E. L. Lehmann. Basic Concepts of Probability and Statistics, 2nd Ed. San Francisco: Holden-Day, 1970.

16. Hogg, R. V., and A. T. Craig. Introduction to Mathematical Statistics, 3rd Ed. London: The Macmillan Company, 1970.

17. Kendall, M. F., and A. Stuart. The Advanced Theory of Statistics, Vol. 2. New York: Hafner Publishing Company, 1961.

18. Kenney, J. F., and E. S. Keeping. Mathematics of Statistics, Part II, 2nd Ed. New York: D. Van Nostrand Company, 1941.

19. Leslie, P. H. "A Simple Method of Calculating the Exact Probability in 2 x 2 Contingency Tables With Small Marginal Totals." Biometrika, 42:522-523 (1955).

20. Leyton, M. K. "Rapid Calculation of Exact Probabilities for 2 x 3 Contingency Tables." Biometrics, 24:714-717 (1968).

21. Lieberman, G. J., and D. B. Owen. Tables of the Hypergeometric Probability Distribution. Stanford: Stanford University Press, 1961.

22. McCracken, D. D. A Guide to FORTRAN IV Programming. New York: John Wiley & Sons, 1965.

23. Mendenhall, W. Introduction to Probability and Statistics, 2nd Ed. Belmont: Wadsworth Publishing Company, 1967.

24. Mainland, D., and I. M. Murray. "Tables for Use in Fourfold Contingency Tests." Science, 116:591-594 (1952).

25. Mood, A. M., and F. A. Graybill. Introduction to the Theory of Statistics, 2nd Ed. New York: McGraw-Hill, 1963.

26. Pierce, A. Fundamentals of Nonparametric Statistics. Belmont: Dickenson Publishing Company, 1970.

27. Roy, S. N., and S. K. Mitra. "An Introduction to Some Non-parametric Generalisations of Analysis of Variance and Multivariate Analysis." Biometrika, 43:361-380 (1956).

28. Smelsey, S. O. "Study of Pilots Who Have Made Multiple Ejections." Aerospace Medicine, 41:563-566 (1970).

29. Tocher, K. D. "Extension of the Neyman-Pearson Theory of Tests to Discontinuous Variates." Biometrika, 37:130-139 (1950).

30. Von Mises, R. Mathematical Theory of Probability and Statistics. New York: Academic Press, 1964.

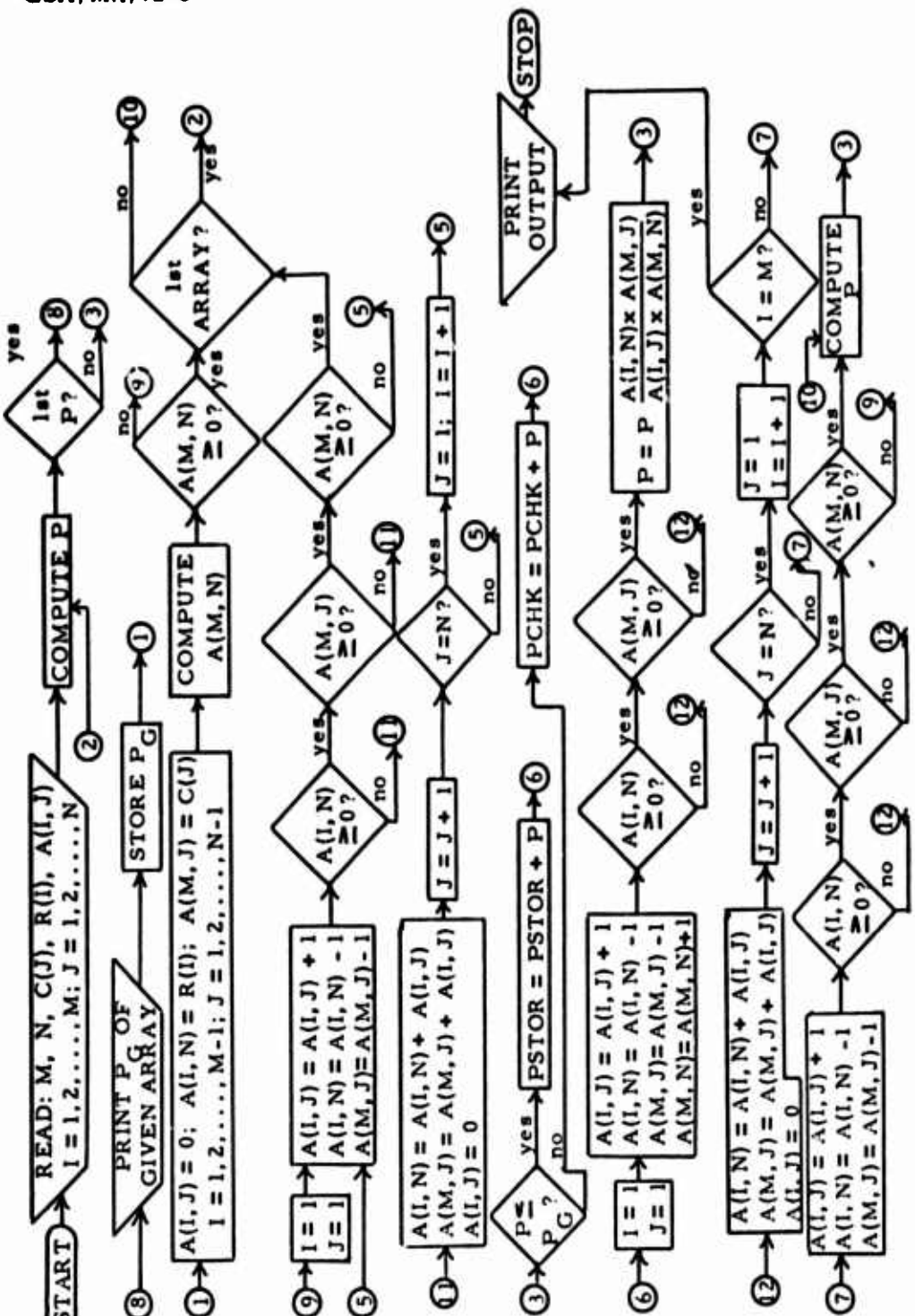
31. Wald, A. "Contributions to the Theory of Statistical Estimation and Testing Hypotheses." The Annals of Mathematical Statistics, 10:299-326 (1939).

32. Wilks, S. S. "The Large-sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses." The Annals of Mathematical Statistics, 9:60-62 (1938).

33. Yates, F. "A Note on the Application of the Combination of Probabilities Test to a Set of 2 x 2 Tables." Biometrika, 42: 157-175 (1955).

Appendix A

Flow Diagram for the Program


START
READ: M, N, C(J), R(I), A(I, J)
I = 1,2,....,M; J = 1,2,....,N
COMPUTE P
1st P?
yes
no
PRINT P_G OF GIVEN ARRAY
STORE P_G
A(I, J) = 0; A(I, N) = R(I); A(M, J) = C(J)
I = 1,2,....,M-1; J = 1,2,....,N-1
COMPUTE A(M, N)
A(M, N) ≥ 0?
1st ARRAY?
I = 1
J = 1
A(I, J) = A(I, J) + 1
A(I, N) = A(I, N) - 1
A(M, J) = A(M, J) - 1
A(I, N) ≥ 0?
A(M, J) ≥ 0?
A(I, N) = A(I, N) + A(I, J)
A(M, J) = A(M, J) + A(I, J)
A(I, J) = 0
J = J + 1
J = N?
J = 1; I = I + 1
P ≤ P_G?
PSTOR = PSTOR + P
PCHK = PCHK + P
P = P A(I, N) x A(M, J) / A(I, J) x A(M, N)
A(M, N) = A(M, N) + 1
I = M?
PRINT OUTPUT
STOP

Appendix B

The Program

```
      N2=1
C INITIALIZE P=THE PROBABILITY
      P=1.
C INITIALIZE D AND THEN INCREMENT UP TO THE TOTAL SAMPLE SIZE.
      D=1.
C IC COUNTS THE NUMBER OF COLUMN TOTALS BEING INCREMENTED.
      IC=1
C C PROVIDES A COUNT FOR THE COLUMN TOTALS.  IT IS INCREMENTED BY ONE
C AFTER EVERY N ITERATIONS UP TO THE TOTAL FOR COLUMN ONE.  THE NUMBER
C OF ITERATIONS BETWEEN EACH INCREMENTATION IS REDUCED BY ONE EACH TIME
C A SUCCEEDING COLUMN TOTAL IS ACHIEVED.
      C=1.
C IR ESTABLISHES POINTS FOR INCREMENTING C
      IR=1
      DO 44 I1=1,M
C "A" IS INCREMENTED UP TO EACH ROW TOTAL
      A=1.
      DO 44 J1=1,N
C "L" CANNOT BE ZERO AS AN INDEXING PARAMETER FOR THE NEXT DO LOOP.
      L=IA(I1,J1)+1
C EACH CELL TOTAL IS INCREASED BY ONE BUT COMPUTATION IS DONE UP TO
C (L-1).  THUS, A ZERO CELL (WHERE ZERO FACTORIAL=ONE) DOES NOT CHANGE
C ANY INTERMEDIATE CALCULATION BUT THE CELL IS ACCOUNTED FOR.
      DO 44 K1=1,L
      IF((L-K1).LE.0)GO TO 44
      AK=K1
      PA=A/AK
      A=A+1.
      P=P*PA
      D=D+1.
   24 IF(P-ASSIGN
      IC=IC+1
      IF((I1-IC).GT.0)GO TO 44
      IR=IR+1
      IC=1
      C=IR
```

```
C SCALE P BY A FACTOR OF E1' SO THAT1 E-1F.LT.P.LE.1.
C SCALING MAY HAVE TO BE DONE AFTER STATEMENT 24 ON SMALL MACHINES.
   30 IF (P.LT.1.E-10)GO TO 31
   32 IF (P.LT.1.)GO TO 33
      P=P*1.E-1
C LARGE IS SCALING FACTOR .  A P .GT. THE GIVEN ARRAY IS STORED IN
C BANK(LARGE) AND A P .LT. THE GIVEN ARRAY IS PUT IN ESTOR(LARGE)
      LARGE=LARGE-1
      GO TO 32
   31 P=P*1.E10
      LARGE=LARGE+1
      GO TO 3?
   ?? IF((IC11-I3).GE.0)GO TO 44
      N?=N?+1
      IF((N-N?).LT.1)GO TO 44
      IC11=IC(N?)
      N1=N1-1
      GO TO 3?
   44 CONTINUE
      MOVE=MOVE+1
C 1ST COMPUTATION USED WITH FUTURE COMPARISONS.  THEN GO TO 150 WITH
C PROBABILITY OF NEXT FEASIBLE ARRAY
      IF(MOVE.EQ.2)GO TO 15?
      POUT=P
      ITEM=LARGE-1
      IF(ITEM.EQ.1)GO TO 62
      DO 61 K=1,ITEM
   61 POUT=POUT*1.E-10
   62 PRINT 54,POUT
   54 FORMAT(3X,'THE PROBABILITY OF THE GIVEN ARRAY IS',/,2X,E25.15)
      TYP P*1.E1?
C INCREMENTS A COMPARISON TO BE USED WITH LAR1 IN CHECKING ALL OTHER
C FEASIBLE ARRAYS WITH THE SAME FIXED MARGINAL TOTALS.
      LAR1=LARGE
      I?=M-1
      I1=N-1
```

```
C DETERMINE THE FIRST FEASIBLE ARRAY HAVING THE SAME FIXED MARGINAL
C TOTALS AS THE GIVEN ARRAY.
C SET ALL CELLS TO ZERO EXCEPT LAST CELL IN EACH ROW AND COLUMN.  THEY
C ARE SET TO THE MARGINAL TOTALS.
      DO 90 I=1,IM
      IX(I)=IR(I)
      DO 80 J=1,IN
   80 IA(I,J)=0
      DO 85 J=1,IN
   85 IY(J)=IC(J)
C DETERMINE IZ=IA(M,N)
   86 IS=0
      DO 90 IFF=1,IM
   90 IS=IS+IX(IFF)
      IZ=IC(N)-IS
   94 IF(IZ.LT.0)GO TO 202
      GO TO (16,600),MOVE
   95 IS=0
      DO 98 IFF=1,IM
   98 IS=IS+IX(IFF)
      IZ=IC(N)-IS
      IF(IZ.GE.0)GO TO 600
  202 IA(1,1)=IA(1,1)+1
      IX(1)=IX(1)-1
      IY(1)=IY(1)-1
      IF(IY(1).LT.0) GO TO 220
      IF(IX(1).LT.0) GO TO 220
      IZ=IZ+1
      GO TO 94
C A NEGATIVE VALUE RESULTED.  RESET TO ZERO THE LAST CELL INCREMENTED
C AND MOVE TO THE NEXT CELL IN THAT ROW.
  220 I3=1
      J3=1
  222 IX(I3)=IX(I3)+IA(I3,J3)
      IY(J3)=IY(J3)+IA(I3,J3)
      IA(I3,J3)=0
```

```
      J3=J3+1
C IF NO CELLS REMAIN IN THE CURRENT ROW GO TO THE FIRST CELL IN THE NEXT
C ROW.
      IF((IM-J3).LT.0)GO TO 240
      IA(I3,J3)=IA(I3,J3)+1
      IY(J3)=IY(J3)-1
      IY(I3)=IX(I3)-1
      IF(IY(J3).LT.0)GO TO 222
      IF(IX(I3))222,86,86
  240 I3=I3+1
      IF((IM-I3).LT.0)GO TO 810
      J3=1
  245 IA(I3,J3)=IA(I3,J3)+1
      IY(J3)=IY(J3)-1
      IY(I3)=IX(I3)-1
C CHECK THAT ALL AFFECTED CELLS ARE POSITIVE.
      IF(IY(J3).LT.0)GO TO 250
      IF(IY(I3).GE.0)GO TO 86
  250 IY(I3)=IX(I3)+IA(I3,J3)
      IY(J3)=IY(J3)+IA(I3,J3)
      IA(I3,J3)=0
      J3=J3+1
      IF (IM-J3)240,245,245
C COMPARATOR SEGMENT
  150 DO 151 JOINT=1,40
      POWY(JOINT)=0.
  151 PSTOR(JOINT)=0.
      LAR2=LARGE
C SCALE WHERE NECESSARY.  ON SMALL MACHINES THIS MAY HAVE TO BE DONE
C IN INTERMEDIATE STEPS DURING THE CALCULATIONS FROM STATEMENT 8810
C TO STATEMENT 8750
  152 IF (P.GT.1.E-10)GO TO 315
      P=P*1.E10
      LARGE=LARGE+1
      LAR2=LARGE
      GO TO 152
```

```
  315 IF (P.LT.1.)GO TO 154
      P=P*1.E-10
      LARGE=LARGE-1
      LAR2=LARGE
      GO TO 315
C COMPARE SCALING FACTORS.
  154 IF (LAR2-LAR1)180,155,164
  155 IMP=P*1.E10
      IF((IMP-INP).GT.0)GO TO 180
C COMPARE PROBABILITIES WITH THE SAME SCALING FACTOR.
  164 PSTOR(LAR2)=PSTOR(LAR2)+P
      ICT=ICT+1
      IF(PSTOR(LAR2).GT.1.)GO TO 165
      GO TO 302
  165 INCR=LAR2-1
      PSTOR(INCR)=PSTOR(INCR)+PSTOR(LAR2)*1.E-10
      PSTOR(LAR2)=0.
      GO TO 302
  180 PCHK(LAR2)=PCHK(LAR2)+P
      ICT=ICT+1
      IF (PCHK(LAR2).GT.1.)GO TO 185
      GO TO 302
  185 INCR=LAR2-1
      PCHK(INCR)=PCHK(INCR)+PCHK(LAR2)*1.E-10
      PCHK(LAR2)=0.
C INCREMENT IA(1,1), ADJUST AFFECTED CELLS, CHECK FOR POSITIVE VALUES.
  302 IA(1,1)=IA(1,1)+1
      IX(1)=IX(1)-1
      IY(1)=IY(1)-1
      IF(IY(1).LT.0)GO TO 320
      IF(IY(1).LT.0)GO TO 320
      A=IY(1)+1
      B=IY(1)+1
      I7=I7+1
      C=IA(1,1)
      D=I7
```

```
      P=P*(A/C)*(B/C)
      GO TO 152
C A NEGATIVE VALUE RESULTED.  STORE OLD VALUES, RESET TO ZERO THE LAST
C CELL INCREMENTED AND MOVE TO THE NEXT CELL IN THAT ROW.
  320 I=1
      J=1
      ITEMPZ=IZ
      ITEMPX(I)=IX(I)+1
  322 ITEMP(I,J)=IA(I,J)-1
      ITEMPY(J)=IY(J)+1
      IX(I)=IX(I)+IA(I,J)
      IY(J)=IY(J)+IA(I,J)
      IA(I,J)=0
      J=J+1
C IF NO CELLS REMAIN IN THE CURRENT ROW GO TO THE FIRST CELL IN THE NEXT
C ROW.
      IF((IM-J).LT.0)GO TO 340
      IA(I,J)=IA(I,J)+1
      IY(J)=IY(J)-1
      IX(I)=IX(I)-1
C CHECK THAT ALL AFFECTED CELLS ARE POSITIVE.
      IF(IY(J).LT.0)GO TO 322
      IF (IX(I))322,356,356
  340 I=I+1
      J=1
      ITEMPX(I)=IX(I)
C IF I.GT. (M-1) NO ADDITIONAL ARRAYS ARE POSSIBLE AND SOLUTION IS
C COMPLETE.
      IF((IM-I).LT.0)GO TO 310
  345 IA(I,J)=IA(I,J)+1
      IY(J)=IY(J)-1
      IX(I)=IX(I)-1
      IF(IY(J).LT.0)GO TO 350
      IF (IX(I).GE.0)GO TO 358
  350 ITEMP(I,J)=IA(I,J)-1
      IY(J)=IY(J)+IA(I,J)
```

```
      DO 649 NT=1,L2
      AP=NU
      P=P*AP
      NU=NU-1
  649 CONTINUE
  650 CONTINUE
      NOD=NOD+1
      I2=1
      GO TO (692,692,695),NOD
  695 NOD=2
      GO TO 701
  700 NOD=1
  701 K4=J
      I5=I
  702 DO 750 KF=1,K4
      GO TO (704,703,705,706),NOD
C ACCOUNT FOR CELLS IN LAST ROW WHICH WAS ALTERED.
  703 NU=ITEMP(I5,KF)
      ND=IA(I5,KF)
      GO TO 708
C ACCOUNT FOR LAST CELL IN EACH COLUMN.  THIS WILL HAVE BEEN COMPLETED
C IF (I.GT.1).
  704 NU=ITEMPY(KF)
      ND=IY(KF)
      GO TO 708
C ACCOUNT FOR LAST CELL IN EACH ROW.
  705 NU=ITEMPX(KF)
      ND=IY(KF)
      GO TO 708
C ACCOUNT FOR I7.
  706 NU=ITEMP7
      ND=I7
  708 IF (NU-ND)710,750,730
  710 L=ND-NU
      DO 720 INT=1,L
      AP=ND
```

```
      P=P/AD
      ND=ND-1
  720 CONTINUE
      GO TO 750
  730 L=NU-ND
      DO 749 NT=1,L
      AD=NU
      P=P*AD
      NU=NU-1
  749 CONTINUE
  750 CONTINUE
      NOD=NOD+1
      GO TO (702,702,755,760,152),NOD
  755 K4=I
      GO TO 702
  760 K4=1
      GO TO 702
  810 DO 805 LAST=2,10
      INCR=10-LAST+1
      INCR1=INCR+1
      PSTOR(INCR)=PSTOR(INCR)+PSTOR(INCR1)*1.E-10
  805 PCHK(INCR)=PCHK(INCR)+PCHK(INCR1)*1.E-10
      RESULT=PSTOR(1)+PCHK(1)
      PRINT 811, PSTOR(1)
      PRINT 812, PCHK(1)
      PRINT 813, RESULT
      PRINT 806, ICT
  806 FORMAT (1X,32HTHE NUMBER OF POSSIBLE ARRAYS IS,I9)
  811 FORMAT (1X, 69HTHE PROBABILITY OF AN ARRAY EQUAL TO OR LESS THAN T
     1HE GIVEN ARRAY IS ,/,20X,E25.15)
  812 FORMAT (1X, 65HTHE PROBABILITY OF AN ARRAY GREATER THAN THE GIVEN
     1ARRAY IS ,/,20X,E25.15)
  813 FORMAT (1X, 51HTOTAL SHOULD EQUAL ONE AS A CHECK. PROGRAM TOTAL IS
     1,/,20X,E25.15)
      STOP
      END
```

VITA

Marvin Francis Schwartz, Jr. was born [redacted] PII Redacted [redacted] He enlisted in the United States Air Force in August 1950. He received a competitive appointment to the United States Military Academy at West Point, N. Y. and entered with the Class of 1956. He graduated 5th in a class of 480 and was commissioned in the United States Air Force. He received his pilot's wings in August 1957 and was assigned to the 11th Troop Carrier Squadron at Dreux Air Base, France. Subsequently he had assignments as: Instructor, Ground Electronics Officer; and, Aide-de-Camp. He flew 593 combat missions as a Forward Air Controller in Vietnam and followed this with duty as a KC-135 Aircraft Commander. Professional military education includes Squadron Officer School and Air Command and Staff College. He is presently assigned in the Graduate Systems Analysis course at the resident school of the Air Force Institute of Technology.

Permanent address: [redacted]

PII Redacted