November 17, 1972

To the Graduate Council:

I am submitting herewith a thesis written by Barbara Jackson Handley entitled "An Examination of Several Methods of Estimation for ABO Blood Groups." I recommend that it be accepted for nine quarter hours of credit in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.

Kimiko O. Bowman
Major Professor

We have read this thesis
and recommend its acceptance:

J. Hull

Donald G. Gardiner

Accepted for the Council:

Hilton A. Smith
Vice Chancellor for
Graduate Studies and Research

AN EXAMINATION OF SEVERAL METHODS OF ESTIMATION FOR ABO BLOOD GROUPS

A Thesis

Presented to

the Graduate Council of

The University of Tennessee

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Barbara Jackson Handley

March 1973

ACKNOWLEDGMENTS

The study described in this thesis was performed while the author was a member of the Applied Science Department of the Computing Technology Center, located at Oak Ridge, Tennessee. This facility is operated for the U. S. Atomic Energy Commission by the Nuclear Division of Union Carbide Corporation.

The author wishes to give much warm thanks to Dr. Kim Bowman, of the Oak Ridge National Laboratory and assistant professor of mathematics at The University of Tennessee, for her advice and encouragement and hard work spent directing this thesis.

Also the author wishes to express her gratitude to Marie Cuthbert and Shirley Gentry for typing the thesis.

1069506

ABSTRACT

The purpose of this study is to investigate the exact distributions of a number of estimators of linkage of the ABO blood group system. Every human being can be classified according to blood type O, A, B, and AB. These four phenotypes are determined by the three alleomorphic genes, A, B, and O, with O recessive to A and B. The basic problem reduces to the estimation of the gene frequencies, p, q, and r, from the observed frequencies of four categories whose probabilities are r^2, $p^2 + 2pr$, $q^2 + 2qr$, and 2pq. Using a large digital computer, the exact distributions of the maximum likelihood, least squares, minimum chi-square, modified chi-square, Hellinger distance, Kullback-Leibler separator, Haldane's discrepancy, Haldane's modified chi-square, Bernstein's, improved Bernstein's, and Wiener's estimators are found for a selection of values for p, q, and r and sample sizes ranging from 30 to 75.

The examination of the exact distributions of p, q, and r reveals that, although the maximum likelihood estimator has many desirable properties, the improved Bernstein's estimator gives comparatively good estimates for p, q, and r for small sample sizes.

TABLE OF CONTENTS

CHAPTER		PAGE
1.	INTRODUCTION	1
2.	FOUNDATIONS OF THEORY OF ESTIMATION	4
3.	METHODS OF ESTIMATION	13
	Least Squares	13
	Maximum Likelihood	14
	Minimum Chi-Square and Associated Methods	15
	Special Methods	18
4.	COMPUTATION AND NUMERICAL METHOD	21
5.	RESULTS AND CONCLUSIONS	27
	Maximum Likelihood	27
	Least Squares	33
	Minimum Chi-Square	35
	Modified Chi-Square	43
	Hellinger's Distance	44
	Kullback-Leibler Separator	49
	Haldane's Discrepancy	53
	Haldane's Modified Chi-Square	59
	Bernstein's Estimator	63
	Improved Bernstein's Estimator	65
	Wiener's Estimator	72
	A Note on Variances	75
	Conclusions	79
	Other Applications	81
BIBLIOGRAPHY		82

CHAPTER PAGE

APPENDIXES 85

A. LISTING OF COMPUTER PROGRAM SYSTEM 86

B. COMPUTER PROGRAM FOR PROBABILITY SPACE FOR MULTINOMIAL DISTRIBUTION OF k CATEGORIES AND n TRIALS 127

VITA 133

LIST OF TABLES

TABLE PAGE

1. Probabilities of Phenotypes and Genotypes in the ABO System 2
2. Parameter Values and Sample Sizes Used in Calculations of Exact Distributions 28
3. Statistics of Maximum Likelihood Estimator for $\bar{p}$ and $\bar{q}$ (First Line) and $\bar{r}$ (Second Line) for p, q, and r Equal to 1/3 and n Increasing from 30 to 50 31
4. Statistics for Maximum Likelihood Estimator with Small r and n Equal to 50 32
5. Statistics of Least Squares Estimator for $\bar{p}$ and $\bar{q}$ (First Line) and $\bar{r}$ (Second Line) with p, q, and r Equal to 1/3 and n from 30 to 75 36
6. Statistics of Minimum Chi-Square Estimator for $\bar{p}$ and $\bar{q}$ (First Line) and $\bar{r}$ (Second Line) with p, q, and r Equal to 1/3 and n from 30 to 50 40
7. Statistics for Minimum Chi-Square Distribution of $\bar{p}$ (First Line), $\bar{q}$ (Second Line), and $\bar{r}$ (Third Line) with Small r and n Equal to 50 41
8. Statistics for Modified Chi-Square Estimator of $\bar{p}$ and $\bar{q}$ (First Line) and $\bar{r}$ (Second Line) for p, q, and r Equal to 1/3 and n from 30 to 50 45
9. Statistics for $\bar{p}$ (First Line), $\bar{q}$ (Second Line), and $\bar{r}$ (Third Line) of Modified Chi-Square Estimator with Small r and n Equal to 50 46

TABLE PAGE

10. Statistics for Hellinger's Distance Estimator for $\bar{p}$ and $\bar{q}$ (First Line) and $\bar{r}$ (Second Line) for p, q, and r Equal to 1/3 and n Varying from 30 to 50 50

11. Statistics for the Kullback-Leibler Separator for $\bar{p}$ and q (First Line) and $\bar{r}$ (Second Line) with p, q, and r Equal to 1/3 and n Varying from 30 to 50 54

12. Statistics for $\bar{p}$ (First Line), $\bar{q}$ (Second Line), and $\bar{r}$ (Third Line) of the Kullback-Leibler Separator Estimator for n Equal to 50 and Large and Small r 55

13. Statistics for Haldane's Discrepancy Estimator for $\bar{p}$ and $\bar{q}$ (First Line) and $\bar{r}$ (Second Line) with p, q, and r Equal to 1/3 and n Varying from 30 to 50 58

14. Statistics for Haldane's Modified Chi-Square Estimator of $\bar{p}$ and $\bar{q}$ (First Line) and $\bar{r}$ (Second Line) with p, q, and r Equal to 1/3 and n from 30 to 50 62

15. Statistics for Bernstein's Estimator for p and q (First Line) and r (Second Line) with p, q, and r Equal to 1/3 and n from 30 to 75 66

16. Statistics for Bernstein's Estimator of $\bar{p}$ (First Line), $\bar{q}$ (Second Line), and $\bar{r}$ (Third Line) for Small r and n Equal to 50 67

17. Statistics for the Improved Bernstein's Estimator for $\bar{p}$ and $\bar{q}$ (First Line) and $\bar{r}$ (Second Line) for p, q, and r Equal to 1/3 and n Varying from 30 to 75 70

TABLE PAGE

18. Statistics for Wiener's Estimator for $\bar{p}$, $\bar{q}$ (First Line) and $\bar{r}$ (Second Line) for p, q, and r Equal to 1/3 and n Varying from 30 to 75 73

19. Variances of the Maximum Likelihood Estimator by Direct Calculation (First Line) and by Exact Distribution (Second Line) for n Equal to 50 and p, q, and r Varying 76

20. Variances of Bernstein's Estimator by Direct Calculation (First Line) and by Exact Distribution (Second Line) for n Equal to 50 and p, q, and r Varying 78

21. Variances of Wiener's Estimator by Direct Calculation (First Line) and by Exact Distribution (Second Line) for n Equal to 50 and p, q, and r Varying 80

22. All Distinct Partitions and Permutations of 3 into 4 Categories 129

23. Structure of Subroutine Space 132

LIST OF FIGURES

FIGURE PAGE

1. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Maximum Likelihood Estimator with p, q, and r Equal to 1/3 and n Equal to 50 34

2. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Least Squares Estimator with p, q, and r Equal to 1/3 and n Equal to 50 37

3. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Minimum Chi-Square Estimator with p, q, and r Equal to 1/3 and n Equal to 50 42

4. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Modified Chi-Square Estimator with p, q, and r Equal to 1/3 and n Equal to 50 47

5. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Hellinger's Distance Estimator with p, q, and r Equal to 1/3 and n Equal to 50 51

6. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Kullback-Leibler Estimator with p, q, and r Equal to 1/3 and n Equal to 50 56

7. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Haldane's Discrepancy Estimator with p, q, and r Equal to 1/3 and n Equal to 50 60

8. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Haldane's Modified Chi-Square Estimator with p, q, and r Equal to 1/3 and n Equal to 50 64

FIGURE PAGE

9. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Bernstein's Estimator with p, q, and r Equal to 1/3 and n Equal to 50 68

10. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Improved Bernstein's Estimator with p, q, and r Equal to 1/3 and n Equal to 50 71

11. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Wiener's Estimator with $\bar{p}$, $\bar{q}$, and $\bar{r}$ Equal to 1/3 and n Equal to 50 74

12. Listing of Subroutine Space 131

1. INTRODUCTION

The purpose of this study is to investigate the exact distributions of a number of estimators of linkage of the ABO blood group system. Every human being can be classified according to blood type O, A, B, and AB. These four phenotypes are determined by the three allomorphic genes A, B, and O, with O recessive to A and B. If the gene frequencies of A, B, and O are denoted by p, q, and r, respectively, with $p + q + r = 1$, then the probabilities of the four phenotypes are as shown in Table 1.

If n individuals are observed with n_1 of type O, n_2 of type A, n_3 of type B, and n_4 of type AB, with $n = \sum n_i$, then the probability of this occurrence is determined by the multinomial distribution shown in equation (1.1).

$$\frac{n!}{n_1!n_2!n_3!n_4!}\,(r^2)^{n_1}(p^2 + 2pr)^{n_2}(q^2 + 2qr)^{n_3}(2pq)^{n_4} \tag{1.1}$$

The problem is to examine the estimators for p, q, and r, given n_1, n_2, n_3, and n_4. Since $p + q + r = 1$, the custom is to estimate p and q and calculate an estimate for r by subtraction. The estimates for p, q, and r are denoted by $\bar{p}$, $\bar{q}$, and $\bar{r}$.

Some standard estimators that will be examined are minimum chi-square, modified chi-square, least squares, maximum likelihood, Hellinger distance, Kullback-Liebler separator, Haldane's discrepancy, and Haldane's modified chi-square. Calculation of estimates for p and q by the method of maximum likelihood is extensively discussed by

TABLE 1. Probabilities of phenotypes and genotypes in the ABO system

Phenotypes	Genotype	Probability
O	OO	r^2
A	AA	p^2
	AO	$2pr$
		$p^2 + 2pr$
B	BB	q^2
	BO	$2qr$
		$q^2 + 2qr$
AB	AB	$2pq$

Kempthorne [1969, pp. 172-178] and Rao [1968, pp. 302-309]. Also some special purpose methods for estimating p, q, and r, Bernstein's method, Bernstein's improved method, and Wiener's method, will be examined. All of the above methods of estimation are referenced and discussed in detail in later sections.

The purpose of this study is to show that, although the maximum likelihood estimators have many desirable properties, the exact distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ reveal many other considerations which are of interest. The extensive calculations necessary for the detailed look at the distribution of $\bar{p}$, $\bar{q}$, and $\bar{r}$ are made possible by the use of large digital computers.

2. FOUNDATIONS OF THEORY OF ESTIMATION

We are considering the estimation of parameters. Let the random variable X have a probability density function which is determinate except for the value of the parameter set $\underline{\Theta}:\{\Theta_1, \Theta_2,\ldots,\Theta_k\}$, which may have any value in a set Ω. This is written as $f(X;\underline{\Theta}),\underline{\Theta}\varepsilon\Omega$, where Ω is the parameter space. A family of probability distribution functions is denoted by $\{f(X;\underline{\Theta}),\underline{\Theta}\varepsilon\Omega\}$; a member of this family is $f(X;\underline{\Theta}),\underline{\Theta}\varepsilon\Omega$. Let $X_1,X_2,\ldots,X_n$ denote a random sample of size n from the distribution $f(X,\underline{\Theta}),\underline{\Theta}\varepsilon\Omega$. The problem is to define an estimate $\underline{T}:\{T_1,T_2,\ldots,T_k\}$ and a function $\underline{u}:\{u_1,u_2,\ldots,u_k\}$ so that $\underline{T}=\underline{u}(X_1,X_2,\ldots,X_n)$ is a "good" estimate of $\underline{\Theta}$. The criteria which determine a "good" estimator will be examined.

There is an important distinction between the method of estimation, which is called the estimator, and the value of the estimator for a certain sample, which is the estimate. Thus the following definition of an estimator is given.

Definition 1: Estimator. An estimator is a function of the observations whose value is an estimate of the parameter $\underline{\Theta}$ of the parent distribution.

The observations are random variables and therefore the estimate can vary from the true value of $\underline{\Theta}$. So a particular estimator cannot be expected to provide a close estimate of $\underline{\Theta}$ for every sample and all values of $\underline{\Theta}$. Criteria are needed by which estimators can be judged that will reflect how they behave in the long run. Thus, the

properties of the distribution of estimates that an estimator generates must be examined. The various properties that a "good" estimator should exhibit will now be defined.

Let $\underline{T}(n):\{T_1(n),T_2(n),\ldots,T_k(n)\}$ be the estimate of $\underline{\Theta}$ for a sample of size n. It is desirable that $\underline{T}(n)$ more accurately estimate $\underline{\Theta}$ as n increases. This leads to the criterion of probability consistency, as it is referred to by Rao [1963, p. 191]. The following definition of probability consistency for several parameters is adapted from Kendall and Stuart [1967, Vol. 2, p. 3].

Definition 2a: Probability consistency. A sequence of estimates given by $\underline{T}(n):\{T_1(n),T_2(n),\ldots,T_k(n)\}$ is consistent for the parameter $\underline{\Theta}$ if and only if, for a given positive ε and δ, there exists an N such that the probability that $|\underline{T}(n) - \underline{\Theta}| < \varepsilon$ is greater than $1 - \delta$ for all $n > N$.

This definition of consistency refers to a limiting property of the estimator and is not valid in connection with a particular sample size. Another definition of consistency is due to Fisher [1956, p. 144], who defines a consistent estimator as ". . . a function of the observed frequencies which takes the exact parametric value when for these frequencies their expectations are substituted". Rao [1962, p. 191] gives a more rigorous definition of this concept of consistency.

Definition 2b is the concept of Fisher consistency, which can be applied to a finite sample size.

Definition 2b: Fisher consistency. A statistic $\underline{T}(n) = \underline{u}(S_n)$, where S_n is the empirical distribution function based on n observations and $\underline{u}$ is weakly continuous functional on space of the distribution functions, is Fisher consistent if and only if $\underline{u}(f(X;\underline{\Theta})) \equiv \underline{\Theta}$.

Fisher consistency is not just a limiting property of a sequence of estimates, as is probability consistency. The definition is applicable to any sample size and may therefore be applied in practical situations. Fisher consistency implies probability consistency.

If an estimator is consistent, its distribution must have an average in a neighborhood of $\underline{\Theta}$ for large samples. It is possible in some cases to require that the distribution of an estimator have a mean value equal to $\underline{\Theta}$ for any sample size n. This leads to the criteria of a bias, as it is given by Hogg and Craig [1967, p. 205].

Definition 3: Bias. An estimate whose mathematical expectation, $E(T_i)$, is equal to the parameter Θ_i is called an unbiased estimate for the parameter Θ_i. Otherwise the estimate is biased.

There may be more than one consistent estimator of $\underline{\Theta}$, even if these are confined to unbiased estimators. Therefore further criteria are needed to choose the better estimate from a set of unbiased, consistent estimators. This can be achieved by considering the variances of the estimators and choosing $\underline{T}$ in such a way that it is not only unbiased but also the variance of T is as small as possible. In general, the estimator with the smaller variance will be distributed more closely around $\underline{\Theta}$, and an unbiased estimator with a small variance will deviate

less from the true value of $\underline{\theta}$ than one with a larger variance. Hence it may be regarded as better. These ideas concerning bias and variance are more formally stated in Definition 4, which is due to Hogg and Craig [1967, p. 205].

Definition 4: Best estimate. For a given sample size n, $\underline{T}(n) = \underline{u}(X_1, X_2, \ldots, X_n)$ is the best estimate of $\underline{\theta}$ if and only if $\underline{T}(n)$ is unbiased and the variance of $\underline{T}(n)$ is less than or equal to the variance of every other unbiased estimate for $\underline{\theta}$.

The above definition states that the criteria of unbiasedness and minimum variance are desirable properties of an estimator. It has been further established that there exists a bound below which the variance of an estimator cannot fall. This leads to the criteria of minimum variance bound, as given in Definition 5. However, before this definition is stated, the concept of information must be developed.

Let $X_1, X_2, \ldots, X_n$ be a random sample from the distribution $f(X;\underline{\theta}), \underline{\theta} \in \Omega$. Then the joint probability distribution of $X_1, X_2, \ldots, X_n$ is called the likelihood function and is written

$$L(X_1 X_2, \ldots, X_n; \underline{\theta}) = f(X_1; \underline{\theta}) f(X_2; \underline{\theta}) \ldots f(X_n; \underline{\theta}) \quad .$$

Let

$$I_{ij} = E[(\partial \log L / \partial \theta_i)(\partial \log L / \partial \theta_j)] \quad .$$

The matrix $[I_{ij}]$, $i,j=1,\ldots,k$ is called the information matrix. An intuitive explanation of what is meant by information is as follows:

the information on the parameter $\underline{\Theta}$ contained in the random variable is the extent to which uncertainty concerning $\underline{\Theta}$ is reduced by the observation of the random variable. The concept of information and a lower bound for the variance of an estimator are related by what is called the Cramér-Rao inequality (see Rao [1968, p. 266]), and is given in Theorem 1.

Theorem 1: Cramér-Rao inequality. If $\underline{T}:\{T_1,T_2,\ldots,T_k\}$ is an unbiased estimate for $\underline{\Theta}$, then the Cramér-Rao inequality states

$$\text{var}(T_i) \geq [I_{ii}]^{-1}$$

where $[I_{ii}]^{-1}$ is the ii^{th} element of the inverse of the information matrix.

Now Definition 5 can be stated as follows.

Definition 5: Minimum variance bound estimator. An estimator which attains the minimum variance bound given in Theorem 1 is called a minimum variance bound estimator. It is stated in Theorem 1 that $\underline{T}$ is an unbiased estimator for $\underline{\Theta}$, then

$$\text{var}(T_i) \geq [I_{ii}]^{-1} \quad .$$

In the case of equality, efficiency can be defined as follows.

Definition 6a: Efficiency. Let $\underline{T}$ be an unbiased estimator for $\underline{\Theta}$. Then $\underline{T}$ is an efficient estimator for $\underline{\Theta}$ if and only if the variance of $\underline{T}$ attains the Cramér-Rao lower bound.

This definition is not entirely satisfactory, because in some cases there are sharper lower bounds. An example is given by Kendall and Stuart [1967, Vol. 2, p. 12]. Thus, it seems preferable to define a limiting property, which leads to the concept of asymptotic efficiency as given by Rao [1968, p. 283].

Definition 6b: Asymptotic efficiency. Let $X_1, X_2, \ldots, X_n$ be a sample of size n, and let $\underline{T}(n)$ be the estimator based on this sample. Then $\underline{T}(n)$ is asymptotically efficient for $\underline{\Theta}$ if and only if

$$\lim E(\underline{T}(n)) = \underline{\Theta}$$

and

$$\lim |\text{var } (T_i(n) - [I_{ii}]^{-1}| = \underline{0}$$

simultaneously as n approaches infinity.

From the developments as given above, a measure of efficiency can be set up to compare estimators, as given in Definition 6c.

Definition 6c: Measure of efficiency. Let $\underline{T}(n)$ be an estimator (based on a sample size of n) of $\underline{\Theta}$. Then the ratio of the Cramér-Rao lower bound to the variance of $\underline{T}(n)$ is the measure of efficiency of $\underline{T}(n)$. The measure of asymptotic efficiency of $\underline{T}(n)$ is the limit of this ratio as n tends to infinity.

This concept is discussed by Fisher ([1925, p. 703] and [1970, p. 132]).

A new definition of asymptotic efficiency is given by Rao ([1961, p. 531] and [1963, p. 193]).

Definition 6d: New definition of asymptotic efficiency. Let $X_1, X_2, \ldots, X_n$ be a random sample from the probability density function $f(X, \underline{\Theta}), \underline{\Theta} \varepsilon \Omega$. Let L be the likelihood function of $\underline{\Theta}$ given the sample, and let

$$\underline{Z}_n = \frac{1}{n} \frac{d \log L}{d\underline{\Theta}} \quad .$$

A consistent estimator $\underline{T}(n)$ (based on the sample of size n) is efficient if and only if

$$\lim(\sqrt{n}\, \underline{T}(n) - \underline{\Theta} - \underline{\beta}(\underline{\Theta})\underline{Z}_n |) = \underline{0} \quad \text{as n approaches infinity}$$

with probability 1 where β does not depend on the X_i.

The criterion of asymptotic efficiency is not decisive in choosing one estimator over another, since different methods of estimation may agree as the sample size tends to infinity, yet differ for finite samples. It is desirable that an estimator summarize all of the information about $\underline{\Theta}$ contained in the sample. This concept is the criterion of sufficiency, introduced by Fisher ([1921, pp. 330-332] and [1925, p. 712]).

Definition 7: Sufficiency. Let $X_1, X_2, \ldots, X_n$ denote a random sample of size n from the probability density function $f(X; \underline{\Theta})$. Let $T_1, T_2, \ldots, T_n$ be n independent estimators of $\Theta_1, \Theta_2, \ldots, \Theta_k$, with each of the Θ_i associated with T_i and $n > k$. Let $g(T_1, T_2, \ldots, T_n; \underline{\Theta})$ be the joint probability density function of $T_1, T_2, \ldots, T_n$, and let $g_k(T_1, T_2, \ldots, T_k; \underline{\Theta})$ be

the joint probability density function of $T_1,T_2,\ldots,T_k$. Then $\underline{T}:\{T_1,T_2,\ldots,T_k\}$ is said to be a vector of joint sufficient estimators for $\underline{\Theta}:\{\Theta_1,\Theta_2,\ldots,\Theta_k\}$ if and only if the conditional probability density function

$$h(T_{k+1},\ldots,T_n|T_1,T_2,\ldots,T_k) = \frac{g(T_1,T_2,\ldots,T_n;\underline{\Theta})}{g_k(T_1,T_2,\ldots,T_k;\underline{\Theta})}$$

does not depend on $\underline{\Theta}$.

The above definition does not say anything about the existence of a sufficient estimator. The Fisher-Neyman criterion, discussed by Hogg and Craig [1967, pp. 211-227], reduces the establishment of the existence of a sufficient estimator to a condition on the likelihood function and is given as follows.

Theorem 2: Fisher-Neyman criterion. Let $X_1,X_2,\ldots,X_n$ denote a random sample from a probability density function $f(X;\underline{\Theta}),\underline{\Theta}\varepsilon\Omega$. Then $\underline{T}:\{T_1,T_2,\ldots,T_k\}$ is a vector of joint sufficient estimators for $\underline{\Theta}:\{\Theta_1,\Theta_2,\ldots,\Theta_k\}$ if and only if

$$f(X_1;\underline{\Theta})f(X_2;\underline{\Theta})\ldots f(X_n;\underline{\Theta}) = L(X_1,X_2,\ldots,X_n;\underline{\Theta}) = g(\underline{T};\underline{\Theta})h(X_1,X_2,\ldots,X_n)$$

where $g(\underline{T};\underline{\Theta})$ is the joint probability density function of $T_1,T_2,\ldots,T_k$ and $h(X_1,X_2,\ldots,X_n)$ does not depend on $\underline{\Theta}$.

One drawback to the Fisher-Neyman criterion is that the probability density function of $\underline{T}$, $g(\underline{T};\underline{\Theta})$, may be tedius to calculate. In

this case a more general theorem, the factorization theorem (see Hogg and Craig [1967, pp. 213-227]) can be used and is given as follows.

Theorem 3: Factorization theorem. Let $X_1, X_2, \ldots, X_n$ denote a random sample from a distribution with probability density function $f(X;\underline{\theta}), \underline{\theta} \in \Omega$. Then $\underline{T}:\{T_1, T_2, \ldots, T_k\}$ is a vector of jointly sufficient estimates of $\underline{\theta}:\{\theta_1, \theta_2, \ldots, \theta_k\}$ if and only if there exist two nonnegative functions h_1 and h_2 such that

$$f(X_1;\underline{\theta})f(X_2;\underline{\theta})\ldots f(X_n;\underline{\theta}) = h_1(\underline{T};\underline{\theta})h_2(X_1, X_2, \ldots, X_n)$$

where $h_2(X_1, X_2, \ldots, X_n)$ does not depend on $\underline{\theta}$.

The discussion of criteria of estimators in this section points out some of the difficulties in finding a satisfactory estimator. A researcher must choose carefully the method of estimation for a particular problem. In the next section, a number of well-known estimators will be described. The object of this study is to examine the exact distributions of these estimators.

3. METHODS OF ESTIMATION

In the previous section, the properties of estimators were discussed. In this section, some methods which have been established to have some of these properties will be examined. Included are general purpose methods and some estimators which have been specifically designed for use in problems in ABO blood group estimation.

Least squares

Suppose that $X_1, X_2, \ldots, X_n$ is a vector of random variables and that the expectation of each is known, so that

$$X_i = E(X_i) + \varepsilon_i$$

where $E(X_i)$ is the expectation of X_i and ε_i is the difference of X_i from its expected value. The expected value, $E(X_i)$, is usually a function of the parameters $\underline{\theta}:\{\theta_1, \theta_2, \ldots, \theta_k\}$. The estimate of $\underline{\theta}$ is obtained by minimizing the expression in equation (3.1) with respect to $\underline{\theta}$.

$$\sum_{i=1}^{n} [X_i - E(X_i)]^2 \tag{3.1}$$

The least squares method can be applied whenever the expectation of the observations in terms of the parameters is known. It is not necessary to know the parent distribution of the observations, as in the case of maximum likelihood.

If the expected values are known linear functions of the parameters, then the least squares estimates are unbiased. For a proof, see

Kendall and Stuart [1967, p. 78]. If, in addition, the deviations, ε_i, are uncorrelated and have zero means and the same variance, then the least squares estimates are unbiased with minimum variance. Anderson [1962, p. 266] discusses the extent to which the assumption of a linear relationship between the expected values and the parameters can be omitted from the above statements.

In the case of normally distributed observations, the least squares estimates coincide with the maximum likelihood estimates.

Maximum likelihood

Let $\underline{X}:\{X_1,X_2,\ldots,X_n\}$ be a sample of random variables. The joint probability of the observations is a function of the parameters $\underline{\Theta}:\{\Theta_1,\Theta_2,\ldots,\Theta_k\}$ and is called the likelihood function, given in equation (3.2).

$$L(\underline{X};\underline{\Theta}) = f(X_1;\underline{\Theta})f(X_2;\underline{\Theta})\ldots f(X_n;\underline{\Theta}) \tag{3.2}$$

The maximum likelihood estimate $\overline{\underline{\Theta}}:\{\overline{\Theta}_1,\overline{\Theta}_2,\ldots,\overline{\Theta}_k\}$ is that value which maximizes the likelihood function as $\underline{\Theta}$ ranges over all possible values. This can be written as equation (3.3).

$$L(\underline{X}:\overline{\underline{\Theta}}) = \sup_{\Theta\epsilon\Omega} L(\underline{X};\underline{\Theta}) \tag{3.3}$$

If the supremum cannot be obtained, a "near maximum likelihood" estimate is defined as in equation (3.4), where $0 < c < 1$, as given by Rao [1968, p. 289].

$$(3.4) \qquad L(X;\overline{\Theta}) \geq c \sup_{\underline{\Theta}\in\Omega} L(\underline{X};\underline{\Theta})$$

In practice, it is simpler to maximize the logarithm of the likelihood function, given in equation (3.5).

$$(3.5) \quad \log L(\underline{X};\underline{\Theta}) = \log f(X_1;\underline{\Theta}) + \log f(X_2;\underline{\Theta}) +\ldots+ \log f(X_n;\underline{\Theta})$$

If a sufficient estimate $\underline{T}:\{T_1,T_2,\ldots,T_k\}$ exists for $\underline{\Theta}$, then the maximum likelihood estimator is a function of it. If $\underline{T}$ is sufficient for $\underline{\Theta}$, then the likelihood function can be written as in equation (3.6).

$$(3.6) \quad L(X_1,X_2,\ldots,X_n;\underline{\Theta}) = g(T_1,T_2,\ldots,T_k;\underline{\Theta})\ h(X_1,X_2,\ldots,X_n)$$

In equation (3.6), $g(\underline{T};\underline{\Theta})$ is the joint probability distribution of $T_1,T_2,\ldots,T_k$ and $h(\underline{X})$ is independent of $\underline{\Theta}$. Then the maximum likelihood estimate is obtained by maximizing $g(T_1,T_2,\ldots,T_k;\underline{\Theta})$.

If a minimum variance bound estimator $\underline{T}$ exists, then it is given by the maximum likelihood method. In general the maximum likelihood estimate is consistent (see Kendall and Stuart [1967, pp. 39-40]) but is usually not unbiased. In nearly all cases where the range of $f(\underline{X}:\underline{\Theta})$ does not depend on $\underline{\Theta}$, the maximum likelihood estimate is asymptotically normally distributed and efficient.

Minimum chi-square and associated methods

If the observations are frequencies of mutually exclusive events or continuous measurements that are grouped into class intervals, the

minimum chi-square methods can be used (see Rao [1968, p. 288]). As in the case of the least squares estimator, these methods do not require the specification of the distribution of the observations. The cell frequencies, $n_1, n_2, \ldots, n_j$, and the respective probabilities, $\pi_1, \pi_2, \ldots, \pi_j$, and the expected values, $n\pi_1, n\pi_2, \ldots, n\pi_j$, where $n = \sum_{i=1}^{j} n_i$, are all that is required. The probabilities, π_i, are functions of the parameter $\underline{\Theta}: \{\Theta_1, \Theta_2, \ldots, \Theta_k\}$. The idea behind the minimum chi-square estimators is to define a measure of the difference between the observed values and the expected values and minimize this value with respect to $\underline{\Theta}$.

The methods that are studied here are the following:

a) Chi-square, given in equation (3.7)

$$\chi^2 = \sum_{i=1}^{j} \frac{(n_i - n\pi_i)^2}{n\pi_i} \tag{3.7}$$

b) Modified chi-square, defined by Neyman [1929], and given in equation (3.8)

$$\text{mod } \chi^2 = \sum_{i=1}^{j} \frac{(n_i - n\pi_i)^2}{n_i} \tag{3.8}$$

where n_i is replaced by 1 if its value is 0.

c) Hellinger distance, given in equation (3.9)

$$\text{H.D.} = \text{arcosine}\left[\sum_{i=1}^{j} \sqrt{(n_i/n)\pi_i}\right] \tag{3.9}$$

d) Kullback-Leibler separator, given in equation (3.10)

$$\text{(3.10)} \qquad \text{K.L.S.} = \sum_{i=1}^{j} \pi_i \log\left(\frac{\pi_i}{n_i/n}\right)$$

e) Haldane's discrepancy, given in equations (3.11) and (3.12)

$$\text{(3.11)} \qquad D_k = \frac{(n+k)!}{n!} \sum_{i=1}^{j} \frac{n_i! \pi_i^{k+1}}{(n_i+k)!} \qquad k \neq -1$$

$$\text{(3.12)} \qquad D_{-1} = \frac{-1}{n} \sum_{i=1}^{j} n_i \log(\pi_i)$$

f) Haldane's modified chi-square, given in equation (3.13)

$$\text{(3.13)} \qquad \text{H. mod } \chi^2 = \sum_{i=1}^{j} \frac{(n_i - n\pi_i)^2}{n_i + 2} \quad .$$

Chi-square, modified chi-square, Hellinger distance, Kullback-Leibler separator, and Haldane's discrepancy are as they appear in Rao [1968, pp. 228-289], and Haldane's modified chi-square is as it appears in Haldane [1955, p. 265].

All of these methods are consistent and efficient under suitable conditions, such as continuous partial derivatives with respect to $\underline{\Theta}$ and continuous $[I_{ij}]$ (Fisher's information). Rao [1961, pp. 531-554] discusses the efficiency of these estimators in relation to maximum likelihood estimator, and in a later paper [1963, pp. 189-206] discusses efficiency and bias of chi-square estimators and maximum

likelihood estimator. Neyman [1949, pp. 239-273] discusses the consistency, asymptotic normality, and efficiency of the chi-square and modified chi-square estimators.

Special methods

All of the previously mentioned methods of estimation, such as maximum likelihood and minimum chi-square, can be used in general for many problems. However, there are methods available that are specifically designed for estimating parameters of the ABO blood groups. These methods are far easier for the experimenter to calculate than the general purpose methods.

If p, q, and r are the true probabilities of the triple allelomorph system ABO, then

$$p + q + r = 1 \quad .$$

Let $\bar{p}$, $\bar{q}$, and $\bar{r}$ denote the estimated gene probabilities from an observed sample. The four phenotypes are represented as O, A, B and AB; and their frequencies are represented as n_1, n_2, n_3, and n_4, respectively, with

$$n = \sum_{i=1}^{4} n_i \quad .$$

Bernstein's method for estimating p, q, and r [1925, pp. 237-270] is given in equations (3.14), (3.15), and (3.16).

$$\bar{p} = 1 - \sqrt{\frac{n_1+n_3}{n}} \tag{3.14}$$

$$\bar{q} = 1 - \sqrt{\frac{n_1+n_2}{n}} \tag{3.15}$$

$$\bar{r} = \sqrt{\frac{n_1}{n}} \tag{3.16}$$

In any observed sample, $\bar{p} + \bar{q} + \bar{r}$ will not necessarily add up to unity. The deviation of $\bar{p} + \bar{q} + \bar{r}$ from unity can be represented by equation (3.17).

$$D = \sqrt{\frac{n_1+n_2}{n}} + \sqrt{\frac{n_1+n_3}{N}} - \sqrt{\frac{n_1}{n}} - 1 \tag{3.17}$$

Bernstein's improved estimates are given by equations (3.18), (3.19), and (3.20).

$$\bar{p} = (1 + D/2)\ (1 - \sqrt{\frac{n_1+n_3}{n}}) \tag{3.18}$$

$$\bar{q} = (1 + D/2)\ (1 - \sqrt{\frac{n_1+n_2}{n}}) \tag{3.19}$$

$$\bar{r} = (1 + D/2)\ (\sqrt{\frac{n_1}{n}} + D/2) \tag{3.20}$$

The third estimator to be considered is Wiener's method [1929, p. 479] given by equations (3.21), (3.22), and (3.23).

$$\bar{p} = \sqrt{\frac{n_1+n_2}{n}} - \sqrt{\frac{n_1}{n}} \tag{3.21}$$

$$(3.22) \qquad \bar{q} = \sqrt{\frac{n_1+n_3}{n}} - \sqrt{\frac{n_1}{n}}$$

$$(3.23) \qquad \bar{r} = \sqrt{\frac{n_1}{n}}$$

Bernstein's unadjusted estimator and Wiener's estimator are not fully efficient; but Bernstein's method is generally more efficient than that of Wiener, as shown by Chakraborty [1970, pp. 21-26]. The improved Bernstein method is fully efficient, as shown by Sukhatme [1941, pp. 85-92].

4. COMPUTATION AND NUMERICAL METHOD

The exact distributions of estimators for small samples of about 50 can be found by generating all possible sample configurations (n_1, n_2, n_3, n_4) with $n = \sum n_i$ and calculating the associated probability given in equation (4.1)

$$(4.1) \quad \pi(n_1,n_2,n_3,n_4) = \frac{n!}{n_1!n_2!n_3!n_4!} (r^2)^{n_1}(p^2+2pr)^{n_2}(q^2+2qr)^{n_3}(2pq)^{n_4}$$

given the solution for $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the particular estimator. There are $(n+1)(n+2)(n+3)/6$ unique sample configurations; therefore, it is necessary to divide the solutions for $\bar{p}$, $\bar{q}$, and $\bar{r}$ into, say, 50 intervals such as $0 \leq p \leq .02, \ldots, .98 \leq p \leq 1$ and accumulate the probabilities for each interval. The result is a histogram for each of $\bar{p}$, $\bar{q}$, and $\bar{r}$. Also the statistics of the distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$, such as means, standard deviations, variances, and third and fourth moments, $\sqrt{\beta_1}$ and β_2, and correlations, are calculated.

A Fortran program was written by the author to calculate the statistics and histograms of the estimators and generate the sample configurations. A fast algorithm is necessary for the sample configurations because of the large number of unique solutions. For example, for n equal to 50 the number of configurations is 23,426. The function values of the various estimators were coded in subroutine form and also gradients of the methods when required. A listing of the Fortran system is in Appendix A. A generalized algorithm for generating sample configurations is in Appendix B.

For estimators such as least squares, maximum likelihood, and chi-square methods, the estimator is solved for $\bar{p}$ and $\bar{q}$, and $\bar{r}$ is calculated by subtraction. For example, the least squares estimator is given by equation (4.2).

$$(4.2) \qquad f(\bar{p},\bar{q}) = \sum_{i=1}^{4} [n_i - n\pi_i(\bar{p},\bar{q})]^2 =$$

$$[n_1 - n(1-\bar{p}-\bar{q})^2]^2 + [n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{qp})]^2$$

$$[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]^2 + [n_4 - n(2\overline{pq})]^2$$

The equation is expressed as a function of $\bar{p}$ and $\bar{q}$ by substituting the expression $1 - \bar{p} - \bar{q}$ for $\bar{r}$. The least squares estimate for p and q is obtained by minimizing the expression given in (4.2) with respect to $\bar{p}$ and $\bar{q}$. There is no algebraic expression for the minimum of such a function, so numerical methods are used. A general purpose constrained nonlinear programming subroutine, Keele, written by Westley [1970, pp. 330-339] was used to find the minimum (or maximum in the case of the maximum likelihood estimator) of the estimator function. Keele allows for a set of linear constraints on the variables and requires feasible initial estimates and analytic calculation of the gradient vectors. The constraints on $\bar{p}$ and $\bar{q}$ are given in equations (4.3), (4.4), and (4.5).

(4.3) $$0 \leq \bar{p} \leq 1$$

(4.4) $$0 \leq \bar{q} \leq 1$$

(4.5) $$0 \leq \bar{p} + \bar{q} \leq 1$$

The initial estimates of $\bar{p}$ and $\bar{q}$ used are Bernstein's estimates given in equations (4.6) and (4.7).

(4.6) $$\bar{p} = 1 - \sqrt{\frac{n_1 - n_3}{n}}$$

(4.7) $$\bar{q} = 1 - \sqrt{\frac{n_1 + n_2}{n}}$$

The analytic expression for the partial derivatives of the estimator function with respect to $\bar{p}$ and $\bar{q}$ are required. For the case of the least squares estimator, the gradients are given in equations (4.8) and (4.9).

(4.8) $$\frac{\partial f(\bar{p},\bar{q})}{\partial \bar{p}} = 4(n_1 - n(1 - \bar{p} - \bar{q})^2)n(1 - \bar{p} - \bar{q})$$
$$- 4(n_2 - n(- \bar{p}^2 + 2\bar{p} - 2\overline{pq}))n(1 - \bar{p} - \bar{q})$$
$$+ 4(n_3 - n(- \bar{q}^2 + 2\bar{q} - 2\overline{pq}))n\bar{q}$$
$$- 4(n_4 - n(2\overline{pq}))n\bar{q}$$

$$(4.9)\qquad \frac{\partial f(\bar{p},\bar{q})}{\partial \bar{p}} = 4(n_1 - n(1 - \bar{p} - \bar{q})^2)n(1 - \bar{p} - \bar{q})$$

$$+ 4(n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq}))n\bar{p}$$

$$- 4(n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq}))n(1 - \bar{p} - \bar{q})$$

$$- 4(n_4 - n(2\overline{pq}))n\bar{p}$$

Other methods for minimizing the estimator function are available. One is outlined in Kempthorne [1969, p. 172] and is known as the scoring method. This method seems to work fine in his example, but odd-looking sample configurations present quite a problem. This procedure was coded for the maximum likelihood estimator for the purpose of checking the calculations produced by Keele. In general Keele finds a better maximum. For many sample configurations, the scoring method diverges wildly and no solution is found. Another function minimization program was tested which does not require gradients and does not provide for constraints. The answers were acceptable but this code took much more computer time than Keele. So it is apparent that a sophisticated method is necessary because of the nature of the functions and the need for fast solutions.

In the case of estimators, Bernstein's method, Bernstein's improved method, and Wiener's method, the expressions for $\bar{p}$, $\bar{q}$, and $\bar{r}$ for a given sample configuration are explicit. In these cases, of course, the scoring method or nonlinear programming code is not necessary.

For example, formulas for Bernstein's estimates are repeated in equations (4.10), (4.11), and (4.12).

$$\bar{p} = 1 - \sqrt{\frac{n_1 + n_2}{n}} \tag{4.10}$$

$$\bar{q} = 1 - \sqrt{\frac{n_1 + n_2}{n}} \tag{4.11}$$

$$\bar{r} = \sqrt{\frac{n_1}{n}} \tag{4.12}$$

If n is equal to 4, and the sample configuration is n_1, n_2, and n_3 equal to 0 and n_4 equal to 4, then the estimates would be calculated as given below in equations (4.13), (4.14), and (4.15).

$$\bar{p} = 1 - \sqrt{\frac{0 + 0}{4}} = 1 \tag{4.13}$$

$$\bar{q} = 1 - \sqrt{\frac{0 + 0}{4}} = 1 \tag{4.14}$$

$$\bar{r} = \sqrt{\frac{0}{4}} = 0 \tag{4.15}$$

It is immediately noticed that $\bar{p} + \bar{q} + \bar{r} = 2$. In contrast, for the general purpose estimators $\bar{p} + \bar{q}$ is restricted to be less than or equal to 1, and $\bar{r}$ is calculated by subtraction.

For all estimators in this study, p and q can be interchanged. That is, if p_1, q_1, r_1 are equal to .2, .4, .4, and p_2. q_2. r_2 are equal to .4, .2, .4, the distribution of p_1 and q_2 would be identical as would the distributions of $\bar{q}_1$ and $\bar{q}_2$ and $\bar{r}_1$ and $\bar{r}_2$ for a fixed n.

Further, if p_1, q_1, r_1 are equal to .2, .4, .4, and p_2, q_2, r_2 are equal to .3, .3, .4, the distributions of $\bar{r}_1$ and $\bar{r}_2$ would not be the same; and distributions of $\bar{p}_2$ and $\bar{q}_2$ would be the same.

All calculations were done on an IBM 360/65 computer. The execution time for n equal to 50 varies from about five minutes for the special methods (Bernstein and Wiener estimators) to about 20 minutes for the maximum likelihood and minimum chi-square estimators.

5. RESULTS AND CONCLUSIONS

In this section the distributions of the various estimators are examined. In Table 2, the parameter spaces and sample sizes that were used in the calculations are specified. The sample size, n, increases from 30 to 50 for p, q, and r equal to 1/3 in order to reveal the consistency of the estimator. For n equal to 50, p, q, and r are varied from near 0 to near 1. The parameter values shown in Table 2 were picked arbitrarily to try to have a picture of the parameter space. (The statistics for all of these parameter values for all estimators will not be presented--only those judged pertinent.) Different populations in the world have been shown to have different gene frequencies. For instance, it is pointed out by Li [1965, pp. 134-135] that in England p, q, r are equal to .25, .05, .7; in some American Indian populations p or q is quite small; and among Chinese p, q equal to .24, .21 do not differ widely.

In this section each of the estimators will be examined and then discussed in comparison according to the criteria outlined in Chapter 2.

Maximum likelihood

The likelihood function is given by equation (4.1). The values of $\bar{p}$ and $\bar{q}$ that maximize the likelihood function will also maximize the logarithm of the likelihood function, given in equation (5.1).

TABLE 2. Parameter values and sample sizes used in calculations of exact distributions.

n	p	q	r
30	1/3	1/3	1/3
35	1/3	1/3	1/3
40	1/3	1/3	1/3
45	1/3	1/3	1/3
	.05	.05	.9
	.15	.15	.7
	.25	.25	.5
	1/3	1/3	1/3
	.4	.4	.2
50	.45	.45	.1
	.55	.25	.2
	.65	.15	.2
	.75	.1	.15
	.85	.05	.1

$$(5.1)\qquad \log L(\bar{p},\bar{q}) = \log\left(\frac{n!}{n_1!n_2!n_3!n_4!}\right) + n_1 \log[(1 - \bar{p} - \bar{q})^2]$$

$$+ n_2 \log(-\bar{p}^2 + 2\bar{p} - 2\bar{pq})$$

$$+ n_3 \log(-\bar{q}^2 + 2\bar{q} - 2\bar{pq}) + n_4 \log(2\bar{pq})$$

The partial derivatives of the logarithm of the likelihood function are given in equation (5.2) and (5.3).

$$(5.2)\qquad \frac{\partial \log L(\bar{p},\bar{q})}{\partial \bar{p}} = \frac{-2n_1}{(1-\bar{p}-\bar{q})} + \frac{n_2(-2\bar{p} + 2 - 2\bar{q})}{(-\bar{p}^2 + 2\bar{p} - 2\bar{pq})}$$

$$- \frac{2n_3}{(-\bar{q} + 2 - 2\bar{p})} + \frac{n_4}{\bar{p}}$$

$$(5.3)\qquad \frac{\partial \log L(\bar{p},\bar{q})}{\partial \bar{q}} = \frac{-2n_1}{(1 - \bar{p} - \bar{q})} - \frac{2n_2}{(-\bar{p} + 2 - 2\bar{q})}$$

$$+ \frac{n_3(-2\bar{q} + 2 - 2\bar{p})}{(-\bar{q}^2 + 2\bar{q} - 2\bar{pq})} + \frac{n_4}{\bar{q}}$$

On examining equations (5.1), (5.2), and (5.3), it is seen that if $\bar{p}$ or $\bar{q}$ is equal to 0 or $\bar{p} + \bar{q}$ is equal to 1, a zero denominator and a zero argument to the logarithm will occur. This fact causes the maximum likelihood estimator to be a little more difficult to work with when finding the exact distribution. This complication was dealt with by replacing a zero denominator or argument of the logarithm by 10^{-20}.

The author also tried setting the whole expression to zero if a singularity occurred. For example, $n_4/\bar{p}$ is replaced by zero. In this case, convergence is faster (fewer iterations and less computer time) but the bias on $\bar{p}$, $\bar{q}$, and $\bar{r}$ is much greater. Of course, when the experimenter is working with real-world data, he will not run into the strange sample configurations that cause these difficult convergence problems.

In Table 3 are the statistics of the distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for p, q, and r equal to 1/3 and for values of n increasing from 30 to 50. It is seen that the mean of $\bar{p}$ and $\bar{q}$ is decreasing monotonically toward 1/3 as n increases. Also, the variances are decreasing, $\sqrt{\beta_1}$'s are getting closer to 0, and β_2's are decreasing monotonically toward 3 as n increases. So it is evident that at this point in the parameter space the maximum likelihood estimator is consistent.

In Table 4 the statistics for the distribtuons of the maximum likelihood estimator are given for large p and small r. As p increases from .65 to .85, the bias of $\bar{p}$ increases. Also at these points in the parameter space, the $\sqrt{\beta_1}$'s are not close to 0, and the β_2's are not generally close to 3. So it is apparent that the distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ are not as well behaved in this situation as they are when p, q, and r are equal to 1/3.

The distributions for the rest of the parameter space given by Table 2 are comparable to the case of p, q, and r equal to 1/3. Therefore, it seems that the maximum likelihood estimator is very good except in the case where p or q is large and r is small.

TABLE 3. Statistics of maximum likelihood estimator for $\bar{p}$ and $\bar{q}$ (First line) and $\bar{r}$ (second line) for p, q, and r equal to 1/3 and n increasing from 30 to 50.

n	μ_1'	σ	σ^2	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3373	.0687	.0047	.0875	.0703	.2702	3.160
	.3254	.0770	.0059	-.0797	325.4	-.1744	3.159
35	.3367	.0633	.0040	.0624	.0504	.2455	3.131
	.3266	.0709	.0050	-.0531	326.6	-.1490	3.126
40	.3362	.0591	.0035	.0420	.0379	.2276	3.109
	.3276	.0661	.0044	-.0383	327.6	-.1327	3.099
45	.3359	.0556	.0031	.0363	.0296	.2113	3.094
	.3282	.0621	.0039	-.0284	328.2	-.1187	3.080
50	.3356	.0527	.0028	.0291	.0237	.1992	3.083
	.3288	.0588	.0035	-.0222	328.8	-.1097	3.067

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

TABLE 4. Statistics for maximum likelihood estimator with small r and n equal to 50.

p q r	μ	σ	$\sqrt{\beta_1}$	β_2
.65	.6575	.0371	.2846	3.218
.15	.1515	.0388	.8503	8.354
.20	.1910	.0641	-.2797	3.161
.75	.7618	.0707	.2149	2.987
.10	.1012	.0333	1.815	21.303
.15	.1370	.0668	-.1717	2.648
.85	.8623	.0833	-1.163	6.520
.05	.0501	.0225	.4349	4.589
.10	.0876	.0846	1.761	9.445

In Figure 1 the histograms of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for p, q, and r equal to 1/3 and n equal to 50 are drawn. The histograms are unimodal and appear to be close to normal.

Least squares

The least squares estimator is given in equation (5.4).

$$f(\bar{p},\bar{q}) = [n_1 - n(1 - \bar{p} - \bar{q})^2]^2 + [n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]^2 + [n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]^2 + [n_4 - n(2\overline{pq})]^2 \tag{5.4}$$

This function is to be minimized with respect to $\bar{p}$ and $\bar{q}$. The partial derivatives of $f(\bar{p},\bar{q})$ are given in equations (5.5) and (5.6).

$$\frac{\partial f(\bar{p},\bar{q})}{\partial \bar{p}} = 4[n_1 - n(1 - \bar{p} - \bar{q})^2]\, n(1 - \bar{p} - \bar{q}) - 4[n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]\, n(1 - \bar{p} - \bar{q}) + 4[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]\, n\bar{q} - 4[n_4 - n(2\overline{pq})]\, n\bar{q} \tag{5.5}$$

$$\frac{\partial f(\bar{p},\bar{q})}{\partial \bar{p}} = 4[n_1 - n(1 - \bar{p} - \bar{q})^2]\, n(1 - \bar{p} - \bar{q}) + 4[n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]\, n\bar{p} \tag{5.6}$$

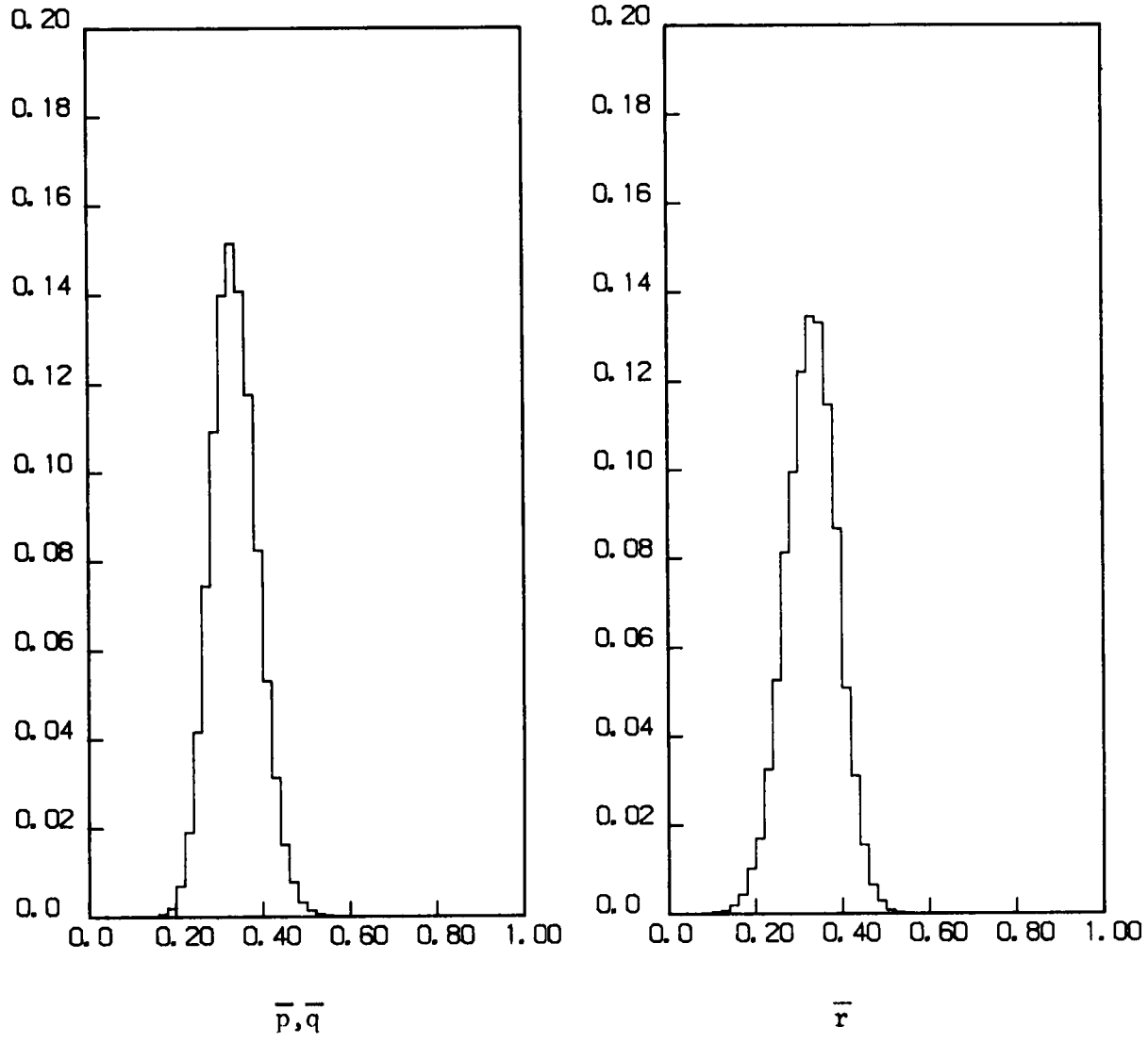


FIGURE 1. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the maximum likelihood estimator with p, q, and r equal to 1/3 and n equal to 50.

$$- 4[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]\, n(1 - \bar{p} - \bar{q})$$

$$- 4[n_4 - n(2\overline{pq})]\, n\bar{p}$$

There are no singularities of these functions in the whole parameter space. The least squares estimator converges faster than any of the other estimators discussed here which use the nonlinear programming technique.

Table 5 gives the statistics of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for p, q, and r equal to 1/3 and n increasing from 30 to 75.

It is seen that the expected value for $\bar{p}$ and $\bar{q}$ decreases monotonically toward 1/3 as n increases. Also, the standard deviations of $\bar{p}$, $\bar{q}$, and $\bar{r}$ are decreasing; and $\sqrt{\beta_1}$'s are getting closer to 0; and β_2's are getting closer to 3.

Figure 2 shows the histograms of $\bar{p}$, $\bar{q}$, and $\bar{r}$ equal to 1/3 and n equal to 50. The distributions are unimodal.

Minimum chi-square

The minimum chi-square estimator is given in equation (5.7).

$$(5.7) \qquad f(\bar{p},\bar{q}) = \frac{[n_1 - n(1 - \bar{p} - \bar{q})^2]^2}{n(1 - \bar{p} - \bar{q})^2} + \frac{[n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]^2}{n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})} + \frac{[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]^2}{n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})} + \frac{[n_4 - n(2\overline{pq})]^2}{n(2\overline{pq})}$$

TABLE 5. Statistics of least squares estimator for $\bar{p}$ and $\bar{q}$ (first line) and $\bar{r}$ (second line) with p, q, and r equal to 1/3 and n from 30 to 75.

n	μ_1'	σ	σ^2	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3403	.0730	.0053	.1330	.0921	.3421	3.248
	.3194	.0884	.0078	-.3957	.3914	-.5721	3.706
35	.3393	.0671	.0045	.0967	.0655	.3201	3.234
	.3213	.0811	.0066	-.3048	321.3	-.5717	3.784
40	.3386	.0623	.0039	.0730	.0487	.3005	3.218
	.3228	.0751	.0056	-.0239	322.8	-.5627	3.821
45	.3380	.0585	.0034	.0568	.0376	.2833	3.202
	.3240	.0703	.0049	-.1910	324.0	-.5497	3.831
50	.3375	.0553	.0031	.0453	.0298	.2680	3.186
	.3249	.0662	.0044	-.1552	324.9	-.5348	3.825
75	.3361	.0445	.0020	.0180	.0123	.2141	3.130
	.3278	.0527	.0028	-.0673	327.8	-.4602	3.714

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

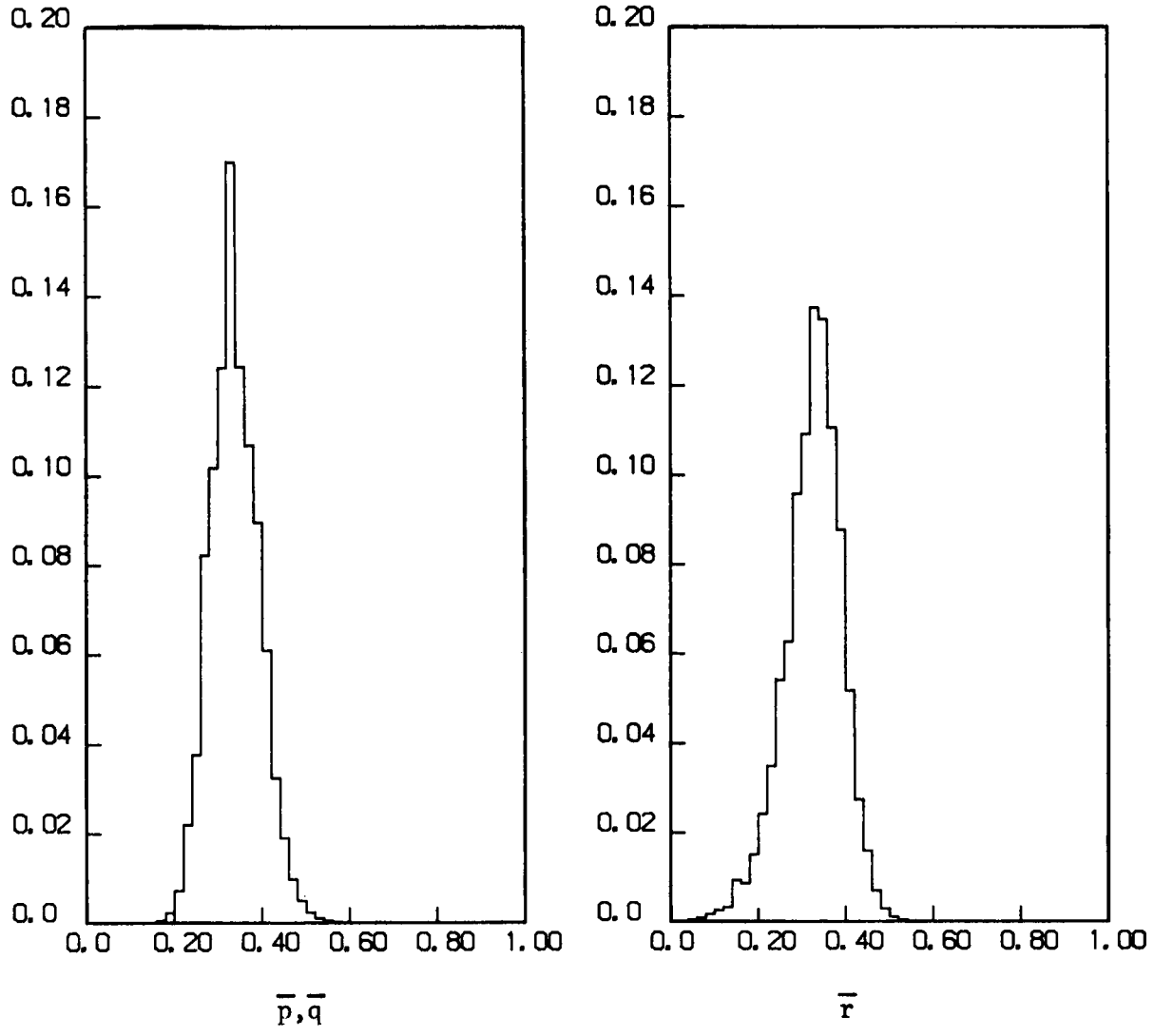


FIGURE 2. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the least squares estimator with p, q, and r equal to 1/3 and n equal to 50.

This function is to be minimized with respect to $\bar{p}$ and $\bar{q}$. The partial derivatives are given in equations (5.8) and (5.9)

(5.8)
$$\frac{\partial f(\bar{p},\bar{q})}{\partial \bar{p}} = \frac{4[n_1 - n(1 - \bar{p} - \bar{q})^2]}{(1 - \bar{p} - \bar{q})} + \frac{2[n_1 - n(1 - \bar{p} - \bar{q})^2]^2}{n(1 - \bar{p} - \bar{q})^3}$$

$$- \frac{4[n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]\ (1 - \bar{p} - \bar{q})}{(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})}$$

$$- \frac{2[n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]^2\ (1 - \bar{p} - \bar{q})}{n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})^2}$$

$$+ \frac{4[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]\ \bar{q}}{(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})}$$

$$+ \frac{2[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]^2\ \bar{q}}{n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})^2}$$

$$- \frac{4[n_4 - n(2\overline{pq})]\ \bar{q}}{(2\overline{pq})} - \frac{2[n_4 - n(2\overline{pq})]^2\ \bar{q}}{n(2\overline{pq})^2}$$

(5.9)
$$\frac{\partial f(\bar{p},\bar{q})}{\partial \bar{q}} = \frac{4[n_1 - n(1 - \bar{p} - \bar{q})^2]}{(1 - \bar{p} - \bar{q})} + \frac{2[n_1 - n(1 - \bar{p} - \bar{q})^2]^2}{n(1 - \bar{p} - \bar{q})^3}$$

$$+ \frac{4[n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]\ \bar{p}}{(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})}$$

$$+ \frac{2[n_2 - (\bar{p}^2 + 2\bar{p} - 2\overline{pq})]^2\ \bar{p}}{n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})^2}$$

$$- \frac{4[n_3 - (-\bar{q}^2 + 2\bar{q} - 2\overline{pq})](1 - \bar{p} - \bar{q})}{(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})}$$

$$- \frac{2[n_3 - (-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]^2 (1 - \bar{p} - \bar{q})}{n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})}$$

$$- \frac{4[n_4 - (2\overline{pq})]\bar{p}}{(2\overline{pq})} - \frac{2[n_4 - 2\overline{pq}]\bar{p}}{n(2\overline{pq})^2}$$

The chi-square estimator is calculated two ways. The first is to delete the whole term if the denominator is zero; the second is to substitute 10^{-20} for a zero expression.

Table 6 gives the statistics of the chi-square estimator for p, q, and r equal to 1/3 and n varying for the two methods of calculation. The statistics in Table 6 show that the distribution of the minimum chi-square estimator becomes more normal with means closer to 1/3 as n increases. The two methods of calculation give comparable statistics.

Table 7 gives the statistics for the minimum chi-square estimator for large values of p and small r for the two methods of calculation. The bias and variances are small, and the values for $\sqrt{\beta_1}$ and β_2 are sometimes far from normal for both methods.

Figure 3 shows the histograms of the distributions of the chi-square estimator calculated by the first method for p, q, and r equal to 1/3 and n equal to 50.

TABLE 6. Statistics of minimum chi-square estimator for $\bar{p}$ and $\bar{q}$ (first line) and $\bar{r}$ (second line) with p, q, and r equal to 1/3 and n from 30 to 50.†

n	μ_1'	σ	σ^2	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3367	.0062	.0044	.0750	.0625	.2580	3.249
	.3266	.0730	.0053	-.0482	326.6	-.1103	3.137
	.3367	.0662	.0044	.0704	.0619	.2426	3.224
	.3266	.0730	.0053	-.0374	326.6	-.0961	3.247
40	.3359	.0575	.0033	.0407	.0340	.2137	3.103
	.3282	.0635	.0040	-.0226	328.1	-.0881	3.055
	.3359	.0575	.0033	.0403	.0343	.2115	3.127
	.3282	.0636	.0040	-.0218*	328.2	-.0850	3.083
50	.3354	.0516	.0027	.0262	.0218	.1907	3.079
	.3291	.0571	.0033	-.0149	329.1	-.0803	3.035
	.3354	.0516	.0027	.0261	.0218	.1903	3.083
	.3291	.0571	.0033	-.0148	329.1	-.0797	3.040

†First method of calculation appears first, with second method below.

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

TABLE 7. Statistics for minimum chi-square distribution of $\bar{p}$ (first line), $\bar{q}$ (second line), and $\bar{r}$ (third line) with small r and n equal to 50.†

p q r	μ_1'	σ	$\sqrt{\beta_1}$	β_2
.65	.6486	.06	.09	5.5
.15	.1534	.04	2.6	54.
.20	.1980	.06	-.32	3.4
	.6485	.07	.14	6.2
	.1532	.04	-2.2	76.
	.1983	.06	.26	11.
.75	.7497	.07	.36	3.9
.10	.1033	.03	1.7	43.
.15	.1470	.07	-.42	2.9
	.7488	.07	.37	3.5
	.1034	.03	.53	52.
	.1478	.07	-.26	4.8

†First method of calculation appears first, with second method below.

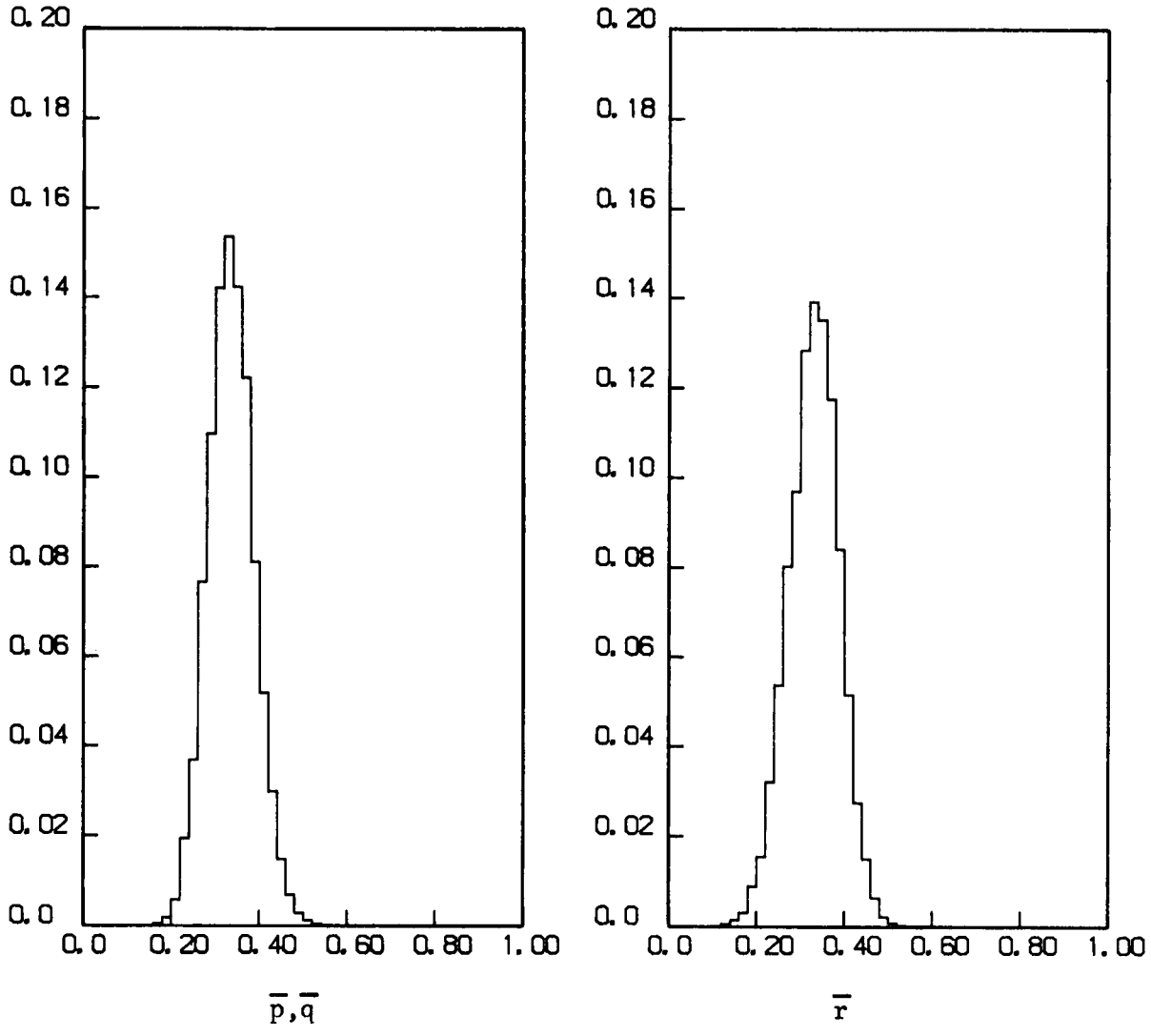


FIGURE 3. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the minimum chi-square estimator with p, q, and r equal to 1/3 and n equal to 50.

Modified chi-square

The modified chi-square estimator is given in equation (5.10).

$$(5.10) \qquad f(p,q) = \frac{[n_1 - n(1 - \bar{p} - \bar{q})^2]^2}{n_1} + \frac{[n_2 + n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]^2}{n_2} + \frac{[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]^2}{n_3} + \frac{[n_4 - n(2\overline{pq})]^2}{n_4}$$

If the denominator, n_i, is 0, it is replaced by 1. The partial derivatives are given in equations (5.11) and (5.12).

$$(5.11) \qquad \frac{\partial f(\bar{p},\bar{q})}{\partial \bar{p}} = \frac{4n[n_1 - n(1 - \bar{p} - \bar{q})^2]\ (1 - \bar{p} - \bar{q})}{n_1} - \frac{4n[n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]\ (1 - \bar{p} - \bar{q})}{n_2} + \frac{4n[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]\ \bar{q}}{n_3} - \frac{4n[n_4 - n(2\overline{pq})]\ \bar{q}}{n_4}$$

$$(5.12) \qquad \frac{f(\bar{p},\bar{q})}{\partial \bar{q}} = \frac{4n[n_1 - n(1 - \bar{p} - \bar{q})^2]\ (1 - \bar{p} - \bar{q})}{n_1}$$

$$+ \frac{4n[n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]\ \bar{p}}{n_1}$$

$$- \frac{4n[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]\ (1 - \bar{p} - \bar{q})}{n_3}$$

$$- \frac{4n[n_4 - n(2\overline{pq})]\ \bar{p}}{n_4}$$

Table 8 gives the statistics for the modified chi-square estimator for p, q, and r equal to 1/3 and n from 30 to 50.

It is seen that the means of p, q, and r are approaching monotonically to 1/3, and the variances are decreasing, and the $\sqrt{\beta_1}$'s and β_2's are close to normal.

Table 9 gives some statistics for $\bar{p}$, $\bar{q}$, and $\bar{r}$ for small r. The figures in Table 9 show that the bias, variance, and $\sqrt{\beta_1}$ for small r are in acceptable ranges, and that β_2 is not far from normal except for very small r (equal to .10).

Figure 4 shows the histograms of the modified chi-square estimator for p, q, and r equal to 1/3 and n equal to 50. The distributions are unimodal and close to symmetric.

The modified chi-square estimator does not have any singularities and therefore is one of the faster converging estimators.

Hellinger's distance

The equation for Hellinger's distance estimator is given in equation (5.13).

TABLE 8. Statistics for modified chi-square estimator of $\bar{p}$ and $\bar{q}$ (first line) and $\bar{r}$ (second line) for p, q, and r equal to 1/3 and n from 30 to 50.

n	μ_1'	σ	σ	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3380	.0762	.0058	.1282	.1211	.29.2	3.593
	.3241	.0845	.0071	-.1797	324.1	-.2975	3.787
35	.3374	.0693	.0048	.0937	.0819	.2811	3.542
	.3252	.0774	.0060	-.1280	325.2	-.2760	3.693
40	.3369	.0639	.0041	.0704	.0580	.2697	3.477
	.3262	.0716	.0051	-.0934	326.2	-.2544	3.600
45	.3364	.0595	.0035	.0539	.0426	.2560	3.406
	.3271	.0668	.0045	-.0692	327.1	-.2325	3.514
50	.3361	.0558	.0031	.0420	.0325	.2411	3.340
	.3279	.0627	.0039	-.0521	327.9	-.2110	3.438

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

TABLE 9. Statistics for $\bar{p}$ (first line), $\bar{q}$ (second line), and $\bar{r}$ (third line) of modified chi-square estimator with small r and n equal to 50.

p q r	μ_1'	σ	$\sqrt{\beta_1}$	β_2
.75	.7546	.05	-.42	3.0
.10	.0978	.03	.34	3.1
.15	.1476	.05	-.33	4.3
.85	.8023	.02	-1.5	7.0
.05	.0526	.02	.32	3.1
.10	.1451	.02	.97	9.9

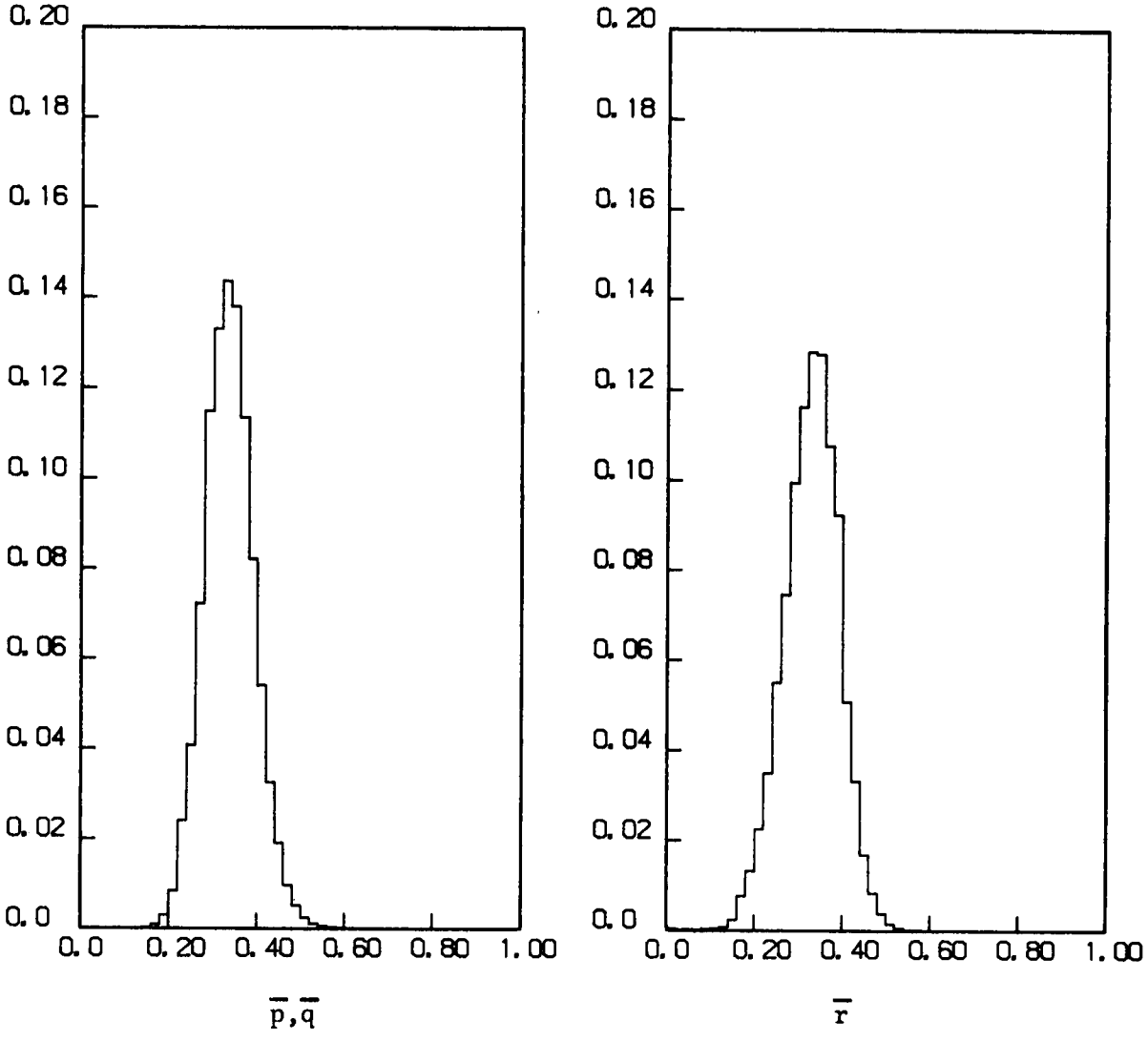


FIGURE 4. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the modified chi-square estimator with p, q, and r equal to 1/3 and n equal to 50.

$$f(\bar{p},\bar{q}) = \text{arcsine}[(n_1/n)^{1/2}\ (1 - \bar{p} - \bar{q}) + (n_2/n)^{1/2}\ (-\bar{p}^2 + 2\bar{p} - 2\overline{pq})^{1/2} + (n_3/n)^{1/2}\ (-\bar{q}^2 + 2\bar{q} - 2\overline{pq})^{1/2} + (n_4/n)^{1/2}\ (2\overline{pq})^{1/2}] \tag{5.13}$$

The partial derivatives with respect to $\bar{p}$ and $\bar{q}$ are given in equations (5.14) and (5.15).

$$\frac{\partial f(\bar{p},\bar{q})}{\partial \bar{p}} = -\ [1 - u^2(\bar{p},\bar{q})]^{-1/2}\ [-\ (n_1/n)^{1/2} + \frac{(n_2/n)^{1/2}\ (1 - \bar{p} - \bar{q})}{(-\bar{p}^2 + 2\bar{p} - 2\bar{q})^{1/2}} - \frac{(n_3/n)^{1/2}\ \bar{q}}{(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})^{1/2}} + \frac{(n_4/n)^{1/2}\ \bar{q}}{(2\overline{pq})^{1/2}}] \tag{5.14}$$

$$\frac{\partial f(\bar{p},\bar{q})}{\partial \bar{q}} = -\ [1 - u^2(\bar{p},\bar{q})]^{-1/2}\ [-\ (n_1/n)^{1/2} - \frac{(n_2/n)^{1/2}\ \bar{p}}{(-\bar{p}^2 + 2\bar{p} - 2\bar{q})^{1/2}} \tag{5.15}$$

$$+ \frac{(n_3/n)^{1/2}\,(1 - \bar{p} - \bar{q})}{(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})^{1/2}}$$

$$+ \frac{(n_4/n)^{1/2}\,\bar{p}}{(2\overline{pq})^{1/2}}]$$

In equations (5.14) and (5.15), $u(\bar{p},\bar{q})$ represents the argument of the arcosine in equation (5.13). The partial derivatives have singularities where the denominator of a term is equal to zero. In this case, the denominator is set to 10^{-20} and calculations continue.

Table 10 gives the statistics for the Hellinger's distance estimator with p, q, and r equal to 1/3 and n varying.

The figures in Table 10 show that the bias and variances are decreasing, and that $\sqrt{\beta_1}$ and β_2 are becoming closer to normal. The $\sqrt{\beta_1}$ and β_2 statistics are close to normal for the rest of the parameter space studied.

Figure 5 shows the histograms of the Hellinger's distance estimator for p, q, and r equal to 1/3 and n equal to 50.

Kullback-Leibler separator

The expression for the Kullback-Leibler separator estimator is given in equation (5.16).

$$(5.16) \qquad f(\bar{p},\bar{q}) = (1 - \bar{p} - \bar{q})^2 \log [(n/n_1)(1 - \bar{p} - \bar{q})^2]$$

$$+ (-\bar{p}^2 + 2\bar{p} - 2\overline{pq}) \log [(n/n_2)(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]$$

TABLE 10. Statistics for Hellinger's distance estimator for $\bar{p}$ and $\bar{q}$ (first line) and $\bar{r}$ (second line) for p, q, and r equal to 1/3 and n varying from 30 to 50.

n	μ_1'	σ	σ^2	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3380	.0707	.0050	.1126	.0821	.3183	3.282
	.3240	.0809	.0065	-.1627	324.0	-.3070	3.398
35	.3371	.0648	.0042	.0769	.0569	.2826	3.228
	.3258	.0736	.0054	-.1054	325.8	-.2642	3.390
40	.3365	.0601	.0036	.0553	.0417	.2540	3.183
	.3270	.0680	.0046	-.0698	327.0	-.2222	3.340
45	.3361	.0564	.0032	.0416	.0319	.2314	3.147
	.3278	.0634	.0040	-.0478	327.8	-.1866	3.278
50	.3358	.0533	.0028	.0324	.0253	.2135	3.120
	.3285	.0598	.0036	.0339	328.5	-.1585	3.219

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

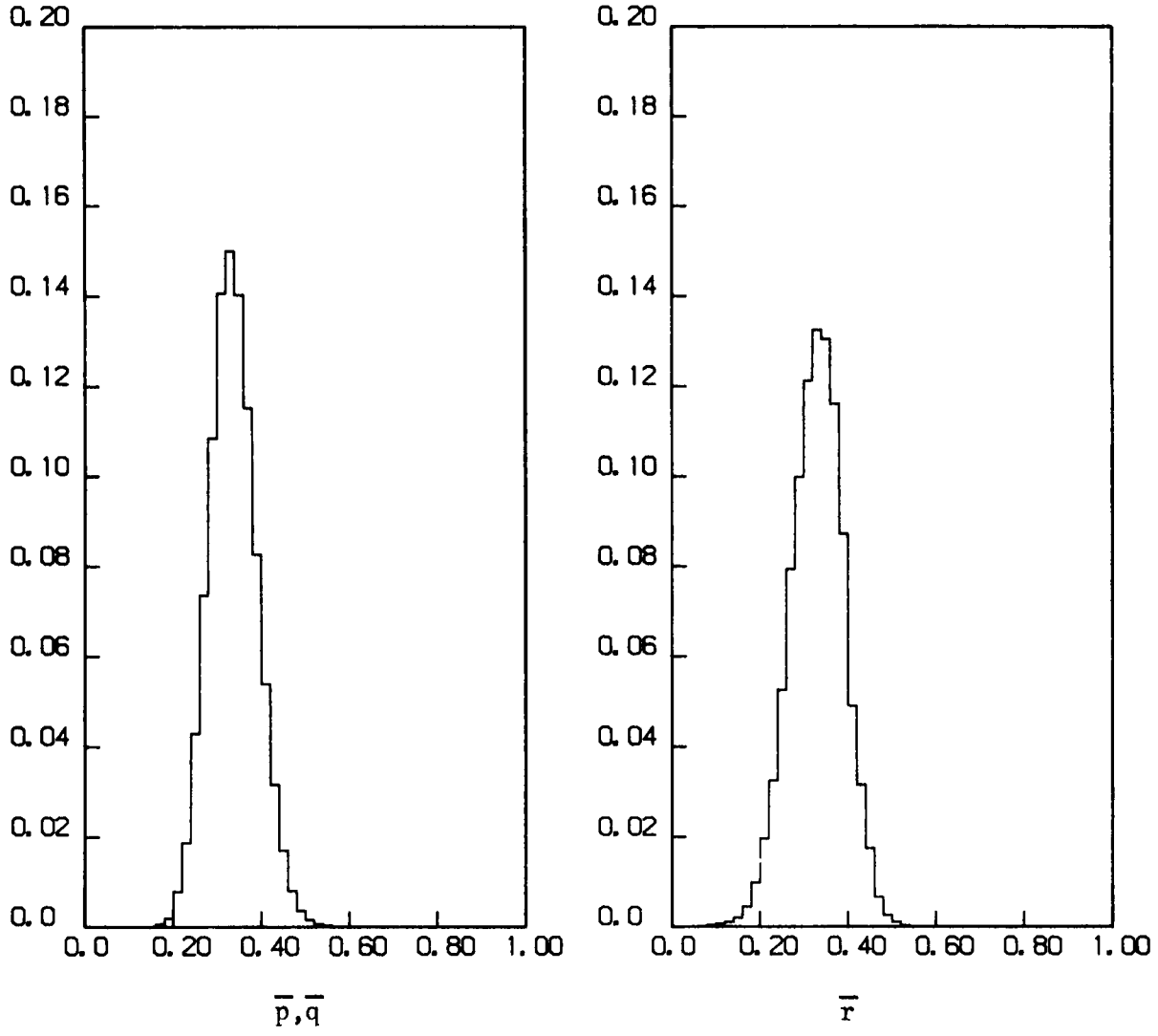


FIGURE 5. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Hellinger's distance estimator with p, q, and r equal to 1/3 and n equal to 50.

$$+ (-\bar{q}^2 + 2\bar{q} - 2\overline{pq}) \log [(n/n_2)(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]$$

$$+ (2pq) \log [(n/n_4)(2\overline{pq})]$$

The partial derivatives with respect to p and q are given in equations (5.17) and (5.18)

$$\frac{\partial f(\bar{p},\bar{q})}{\partial \bar{p}} = - 2(1 - \bar{p} - \bar{q}) \log [(n/n_1)(1 - \bar{p} - \bar{q})^2] \tag{5.17}$$

$$+ 2(1 - \bar{p} - \bar{q}) \log [(n/n_2)(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]$$

$$- 2\bar{q} \log [(n/n_3)(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]$$

$$+ 2\bar{q} \log [(n/n_4)(2\overline{pq})]$$

$$\frac{\partial f(\bar{p},\bar{q})}{\partial \bar{q}} = - 2(1 - \bar{p} - \bar{q}) \log [(n/n_1)(1 - \bar{p} - \bar{q})^2] \tag{5.18}$$

$$- 2\bar{p} \log [(n/n_2)(- \bar{p}^2 + 2\bar{p} - 2\overline{pq})]$$

$$+ 2(1 - \bar{p} - \bar{q}) \log [(n/n_3)(- \bar{q}^2 + 2\bar{q} - 2\overline{pq})]$$

$$+ 2\bar{p} \log [(n/n_4(2\overline{pq})]$$

The estimator and the partial derivatives have singularities where n_i is equal to zero or where the argument to the logarithm is

zero. For calculation purposes, the entire term is set to zero in the case of a singularity.

Table 11 gives the statistics for Kullback-Leibler separator estimator for p, q, and r equal to 1/3 and n varying.

It is interesting to note that the means of $\bar{p}$, $\bar{q}$, and $\bar{r}$ do not monotonically approach 1/3 as the previous estimators discussed have. The bias decreases then increases as n goes from 30 to 50. The variances decrease and $\sqrt{\beta_1}$ and β_2 become more normal as n increases. However, $\sqrt{\beta_1}$ and β_2 for the distribution of $\bar{r}$ are much further from normal than for any of the estimators discussed so far.

Table 12 gives some statistics of the Kullback-Leibler separator for large and small r. The statistics in Table 12 point out some very strange behavior of this estimator. When r is large (equal to .90 and .70), the bias of $\bar{p}$, $\bar{q}$, and $\bar{r}$ is not large, but $\sqrt{\beta_1}$ and β_2 are very far from normal. When r is small (equal to .15 and .10), the bias of $\bar{p}$, $\bar{q}$, and $\bar{r}$ is extremely large and $\sqrt{\beta_1}$ and β_2 are erratic.

Figure 6 shows the histogram for the distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for p, q, and r equal to 1/3 and n equal to 50. The distributions are close to symmetric, although the tails are not smooth.

Haldane's discrepancy

The expression for the Haldane's discrepancy estimator with k equal to 1 is given in equation (5.19).

$$f(\bar{p},\bar{q}) = (n+1)\left[\frac{(1 - \bar{p} - \bar{q})^4}{n_1 + 1} + \frac{(- \bar{p}^2 + 2\bar{p} - 2\overline{pq})^2}{n_2 + 1}\right. \tag{5.19}$$

TABLE 11. Statistics for the Kullback-Leibler separator for $\bar{p}$ and $\bar{q}$ (first line) and $\bar{r}$ (second line) with p, q, and r equal to 1/3 and n varying from 30 to 50.

n	μ_1	σ	σ^2	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3284	.0791	.0063	-.1803	.1556	-.3636	3.970
	.3432	.1085	.0118	2.516	343.2	1.969	9.891
35	.3318	.0764	.0050	-.1008	.1014	-.2883	4.118
	.3364	.0918	.0084	1.485	336.4	1.920	11.57
40	.3336	.0639	.0041	-.0478	.0678	-.1835	4.067
	.3328	.0796	.0063	.8441	332.8	1.670	12.02
45	.3345	.0589	.0035	-.0175	.0472	-.0856	3.921
	.3311	.0709	.0050	.4731	331.1	1.325	11.39
50	.3348	.0550	.0030	-.0008	.0342	-.0050	3.737
	.3302	.0646	.0042	.2594	330.2	.9638	10.01

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

TABLE 12. Statistics for $\bar{p}$ (first line), $\bar{q}$ (second line) and $\bar{r}$ (third line) of the Kullback-Leibler separator estimator for n equal to 50 and large and small r.

p q r	μ_1'	σ	$\sqrt{\beta_1}$	β_2
.05	.0593	.07	0.5	108.
.05	.0593	.07	9.5	108.
.90	.8814	.10	-7.3	61.
.15	.1579	.06	3.3	20.
.15	.1579	.06	3.3	20.
.70	.6841	.10	-4.9	33.
.75	.5244	.27	-.50	1.4
.10	.1122	.14	3.2	13.
.15	.3634	.29	.80	1.8
.85	.4095	.27	.52	1.3
.05	.0686	.09	4.7	30.
.10	.5219	.29	-.38	1.3

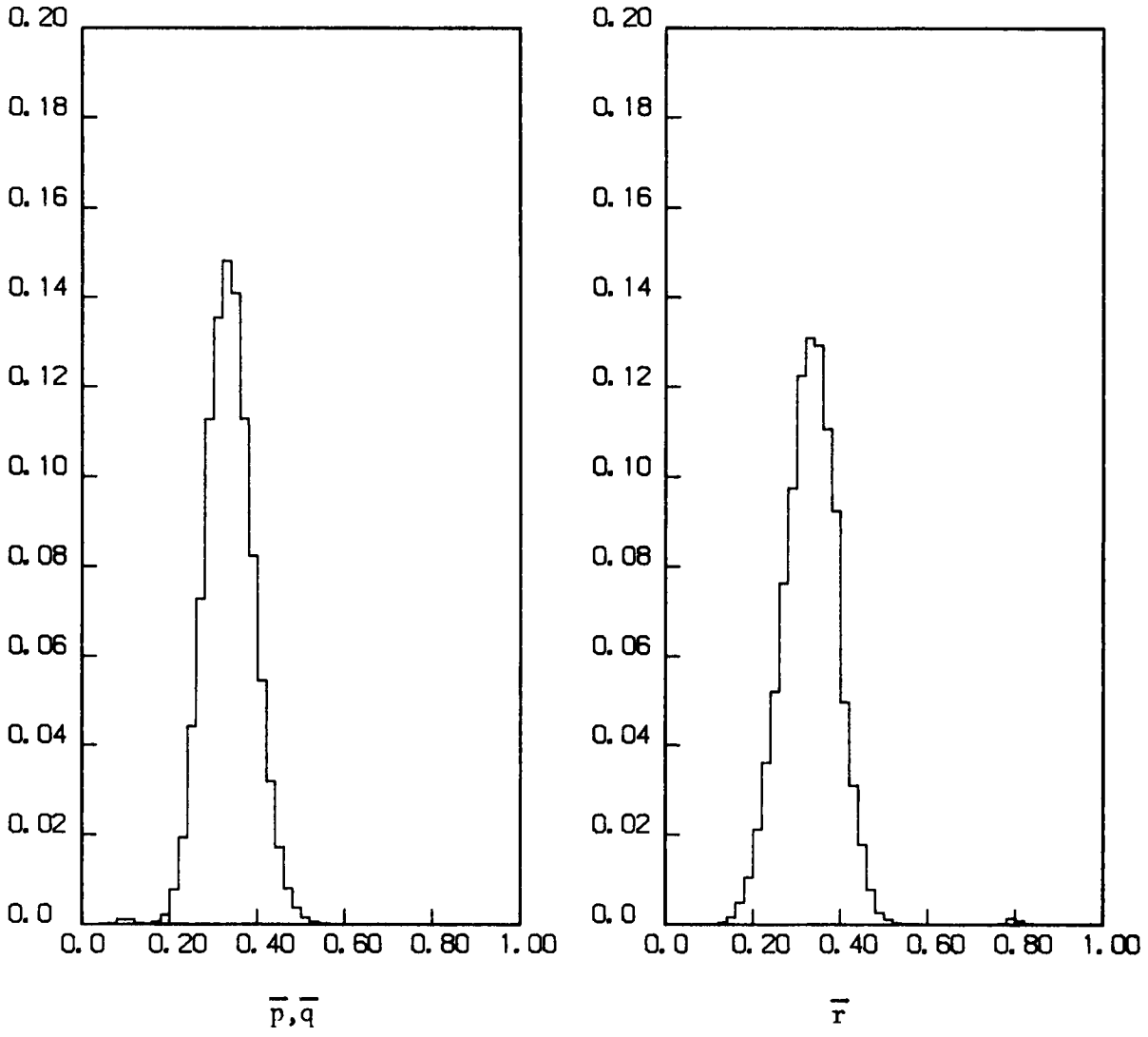


FIGURE 6. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Kullback-Leibler estimator with p, q, and r equal to 1/3 and n equal to 50.

$$+ \frac{(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})^2}{n_3 + 1} + \frac{(2\overline{pq})^2}{n_4 + 1}]$$

The partial derivatives with respect to $\bar{p}$ and $\bar{q}$ are given in equations (5.20) and (5.21).

$$(5.20) \qquad \frac{\partial f(\bar{p},\bar{q})}{\partial \bar{p}} = (n+1)\ [\frac{-4(1 - \bar{p} - \bar{q})^3}{n_1 + 1}$$

$$+ \frac{4(-p^2 + 2\bar{p} - 2\overline{pq})\ (1 - \bar{p} - \bar{q})}{n_2 + 1}$$

$$- \frac{4\bar{q}(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})}{n_3 + 1} + \frac{8\overline{pq}^2}{n_4 + 1}]$$

$$(5.21) \qquad \frac{\partial f(\bar{p},\bar{q})}{\partial \bar{q}} = (n+1)\ [\frac{-4(1 - \bar{p} - \bar{q})^3}{n_1 + 1} - \frac{4\bar{p}(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})}{n_2 + 1}$$

$$+ \frac{4(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})\ (1 - \bar{p} - \bar{q})}{n_3 + 1} + \frac{8\bar{p}^2\bar{q}}{n_4 + 1}\ ]$$

Table 13 gives the statistics for the Haldane's discrepancy estimator for p, q, and r equal to 1/3 and n varying. The figures in Table 13 show that the bias and variances of $\bar{p}$, $\bar{q}$, and $\bar{r}$ are decreasing as n increases and that $\sqrt{\beta_1}$ and β_2 are becoming closer to normal. The $\sqrt{\bar{\beta}_1}$ and β_2 statistics are close to normal for the whole range p, q, and r studied.

TABLE 13. Statistics for Haldane's discrepancy estimator for $\bar{p}$ and $\bar{q}$ (first line) and $\bar{r}$ (second line) with p, q, and r equal to 1/3 and n varying from 30 to 50.

n	μ_1'	σ	σ^2	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3294	.0638	.0041	.0813	.0587	.3125	3.532
	.3412	.0733	.0054	-.1813	341.2	-.4602	4.476
35	.3298	.0594	.0035	.0583	.0427	.2783	3.433
	.3404	.0676	.0046	-.1172	340.4	-.3789	4.246
40	.3302	.0557	.0031	.0433	.0323	.2498	3.343
	.3397	.0631	.0040	-.0772	339.7	-.3078	4.016
45	.3304	.0527	.0028	.0331	.0252	.2262	3.271
	.3391	.0593	.0035	-.0516	339.1	-.2473	3.796
50	.3307	.0501	.0025	.0261	.0203	.2073	3.214
	.3386	.0562	.0032	-.0353	.3386	-.1990	3.605

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

Figure 7 shows the histogram of the Haldane's discrepancy estimator of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for p, q, and r equal to 1/3 and n equal to 50. The distributions are unimodal and close to symmetric.

Haldane's modified chi-square

The equation for the Haldane's modified chi-square estimator is given in equation (5.22).

$$(5.22) \qquad f(\bar{p},\bar{q}) = \frac{[n_1 - n(1 - \bar{p} - \bar{q})^2]^2}{n_1 + 2}$$

$$+ \frac{[n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]^2}{n_2 + 2}$$

$$+ \frac{[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]^2}{n_3 + 2}$$

$$+ \frac{[n_4 - n(2\overline{pq})]^2}{n_4 + 2}$$

The partial derivatives are given in equations (5.23) and (5.24).

$$(5.23) \qquad \frac{\partial f(\bar{p},\bar{q})}{\partial \bar{p}} = \frac{4n[n_1 - n(1 - \bar{p} - \bar{q})^2]^2 (1 - \bar{p} - \bar{q})}{n_1 + 2}$$

$$- \frac{4n[n_2 - n(-\bar{p}^2 + 2\bar{q} - 2\overline{pq})](1 - \bar{p} - \bar{q})}{n_2 + 2}$$

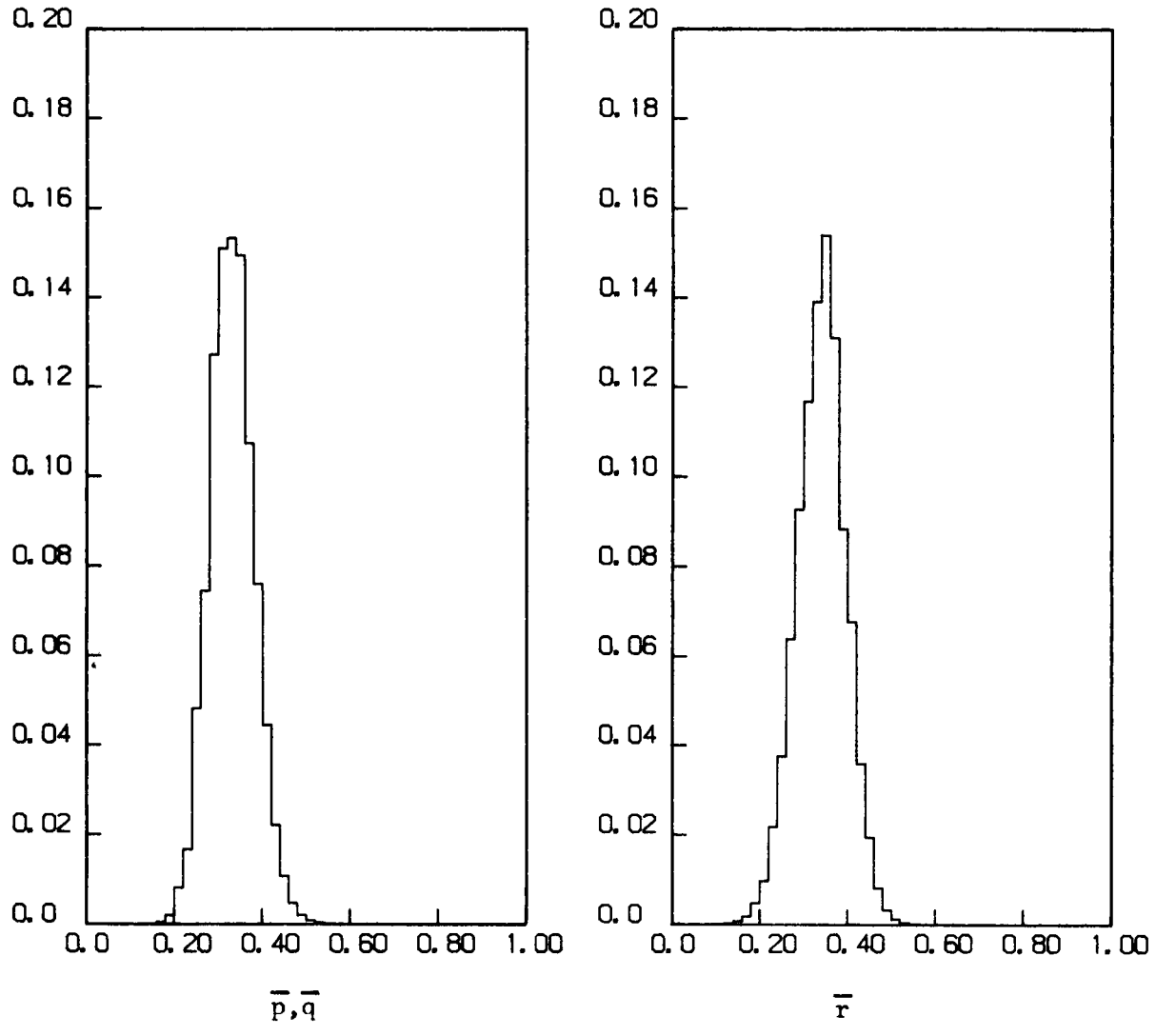


FIGURE 7. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Haldane's discrepancy estimator with p, q, and r equal to 1/3 and n equal to 50.

$$+ \frac{4n\bar{q}[n_3 - n(-\bar{q}^2 + 2\bar{q} - 2\overline{pq})]}{n_3 + 2}$$

$$- \frac{4n\bar{q}[n_4 - n(2\overline{pq})]}{n_4 + 2}$$

(5.24)
$$\frac{\partial f(\bar{p},\bar{q})}{\partial \bar{q}} = \frac{4n[n_1 - n(1 - \bar{p} - \bar{q})^2 (1 - \bar{p} - \bar{q})}{n_1 + 2}$$

$$+ \frac{4n\bar{p}[n_2 - n(-\bar{p}^2 + 2\bar{p} - 2\overline{pq})]}{n_2 + 2}$$

$$- \frac{4n[n_3 - n(-\bar{q}^2 - 2\bar{q} - 2\overline{pq})] (1 - \bar{p} - \bar{q})}{n_3 + 2}$$

$$- \frac{4n\bar{p}[n_4 - n(2\overline{pq})]}{n_4 + 2}$$

Unlike the minimum chi-square estimator, the Haldane's modified chi-square estimator has no singularities. Therefore, the numerical computation of the distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ is faster and less complex.

Table 14 gives the statistics for the Haldane's modified chi-square estimator with p, q, and r equal to 1/3 and n varying. The bias and variances of the distributions decrease as n increases, and the $\sqrt{\beta_1}$ and β_2 statistics become closer to normal. The variances are pretty stable and $\sqrt{\beta_1}$ and β_2 statistics remain close to normal over the whole parameter space studied.

TABLE 14. Statistics for Haldane's modified chi-square estimator of $\bar{p}$ and $\bar{q}$ (first line) and $\bar{r}$ (second line) with p, q, and r equal to 1/3 and n from 30 to 50.

n	μ_1'	σ	σ^2	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3388	.0744	.0055	.1474	.1063	.3581	3.473
	.3234	.0870	.0076	-.3118	322.4	-.4740	3.853
35	.3378	.0679	.0046	.1009	.0723	.3231	3.410
	.3244	.0787	.0062	-.2115	324.4	-.4336	3.888
40	.3370	.0627	.0039	.0721	.0518	.2925	3.352
	.3259	.0722	.0052	-.1445	325.9	-.3834	3.843
45	.3365	.0585	.0034	.0533	.0387	.0266	3.296
	.3270	.0670	.0045	-.0999	327.0	-.3321	3.753
50	.3361	.0551	.0030	.0406	.0299	.2428	3.246
	.3279	.0628	.0039	-.0703	327.9	-.2846	3.646

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

Figure 8 shows the histograms for the distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for p, q, and r equal to 1/3 and n equal to 50. The distributions are unimodal and close to symmetric.

The estimators discussed above are general purpose estimators which can be adapted to many problems. The special purpose estimators, Bernstein's estimator, improved Bernstein's estimator, and Wiener's estimator will now be examined.

Bernstein's estimator

The formulas for Bernstein's estimator are given in equations (5.25), (5.26), and (5.27).

$$\bar{p} = 1 - \sqrt{\frac{n_1 + n_3}{n}} \tag{5.25}$$

$$\bar{q} = 1 - \sqrt{\frac{n_1 + n_2}{n}} \tag{5.26}$$

$$\bar{r} = \sqrt{\frac{n_1}{n}} \tag{5.27}$$

The $\bar{p}$, $\bar{q}$, and $\bar{r}$ are calculated directly; there is no minimization of complicated expressions, no calculations of derivatives. The estimator can easily be calculated by hand by the experimenter, where a method such as minimum chi-square would require a computer program.

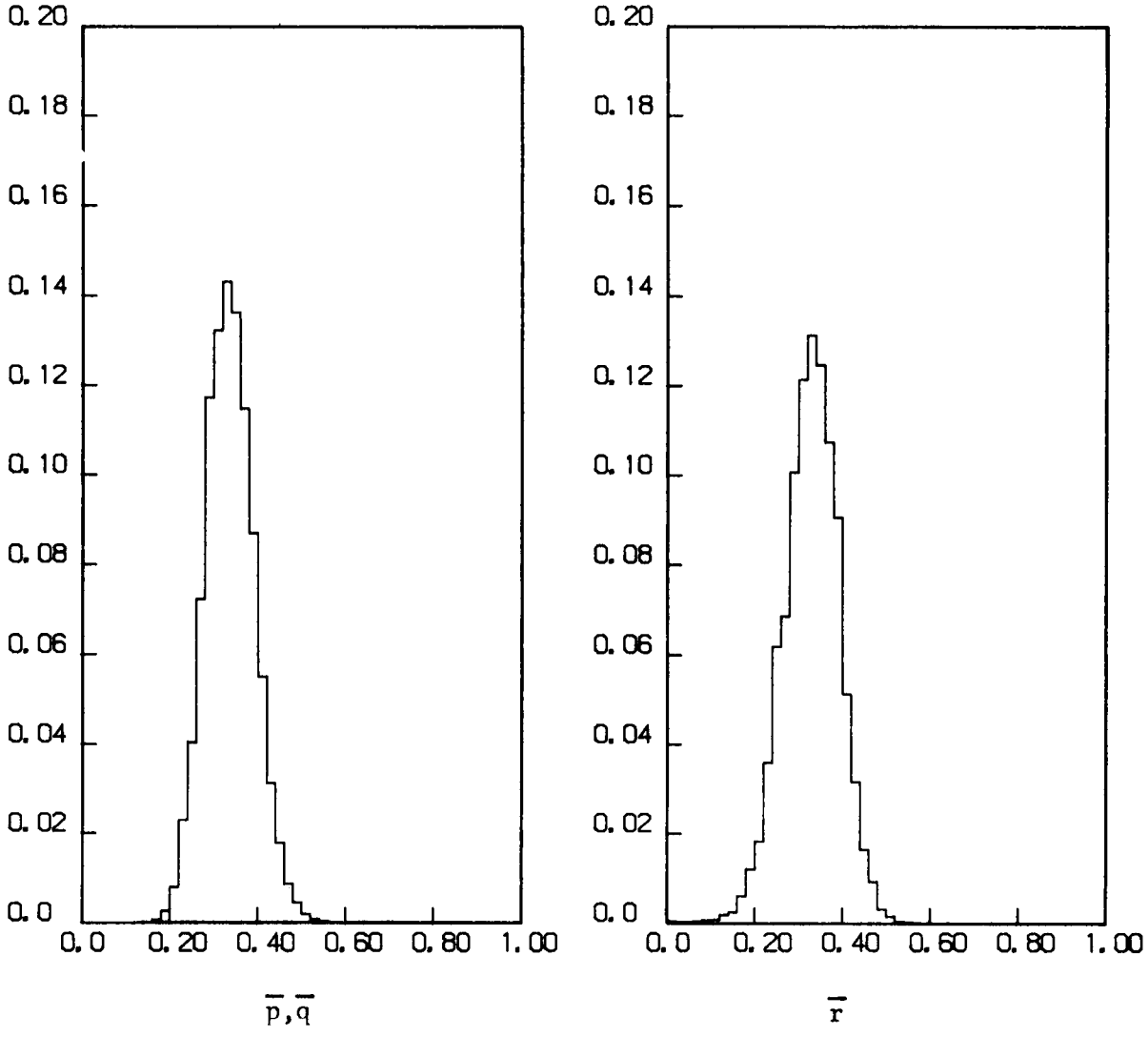


FIGURE 8. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Haldane's modified chi-square estimator with p, q, and r equal to 1/3 and n equal to 50.

Table 15 gives the statistics for the Bernstein estimator for p, q, and r equal to 1/3 and n varying. The figures in Table 15 show that the bias and variances are decreasing as n increases, and that $\sqrt{\beta_1}$ and β_2 are becoming closer to normal.

Table 16 gives some statistics of Bernstein's estimator for small r. The figures in Table 16 show that the bias worsens as r decreases. The variance of $\bar{p}$ increases as p increases, and the variances of $\bar{q}$ and $\bar{r}$ decrease as q and r decrease. The $\sqrt{\beta_1}$ and β_2 are sporadic, but generally worsen as r decreases. Also, it is noticed from Tables 15 and 16 that the sum of $\bar{p}$, $\bar{q}$, and $\bar{r}$ is not equal to 1.

The rest of the parameter space studied, for large r, showed no abnormalities.

The histograms for the distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ of Bernstein's estimator appear in Figure 9. These histograms look very strange after seeing the well-shaped ones representing the general purpose estimators. It is obvious from looking at the equation for $\bar{r}$ that for a given value of n, there are a specific number of solutions over the sample configurations that will be repeated. Thus, for small n, the histogram of $\bar{r}$ will have many modes.

Improved Bernstein's estimator

The formulas for the improved Bernstein's estimator are given in equations (5.28), (5.29), and (5.30).

$$\bar{p} = (1 + d)\left(1 - \sqrt{\frac{n_1 + n_3}{n}}\right) \tag{5.28}$$

TABLE 15. Statistics for Bernstein's estimator for $\bar{p}$ and $\bar{q}$ (first line) and $\bar{r}$ (second line) with p, q, and r equal to 1/3 and n from 30 to 75.

n	μ_1'	σ	σ^2	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3369	.0692	.0048	.0933	.0727	.2812	3.164
	.3185	.0983	.0097	-.0747	318.5	-.7873	4.485
35	.3364	.0639	.0041	.0674	.0524	.2579	3.136
	.3214	.0884	.0078	-.4801	321.4	-.6959	4.491
40	.3360	.0597	.0036	.0509	.0395	.2395	3.116
	.3234	.0809	.0065	-.3166	323.3	-.5980	4.339
45	.3357	.0562	.0032	.0399	.0309	.2246	3.102
	.3248	.0750	.0056	-.2161	324.7	-.5112	4.125
50	.3355	.0532	.0028	.0320	.0248	.2123	3.090
	.3256	.0704	.0050	-.1532	.325.8	-.4394	3.910
75	.3347	.0433	.0019	.0139	.0108	.1714	3.058
	.3286	.0560	.0031	-.0460	328.6	-.2617	3.306

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

TABLE 16. Statistics for Bernstein's estimator of $\bar{p}$ (first line), $\bar{q}$ (second line), and $\bar{r}$ (third line) for small r and n equal to 50.

p q r	μ_1'	σ	$\sqrt{\beta_1}$	β_2
.65	.6569	.0693	.3843	3.718
.15	.1508	.0375	.2743	3.103
.20	.1802	.0868	-.7016	3.113
.75	.7629	.0792	.7833	4.297
.10	.1005	.0310	.3279	3.123
.15	.1201	.0899	-.2245	1.800
.85	.8799	.0899	.2248	1.800
.05	.0503	.0222	.4545	3.229
.10	.0618	.0786	.6264	1.741

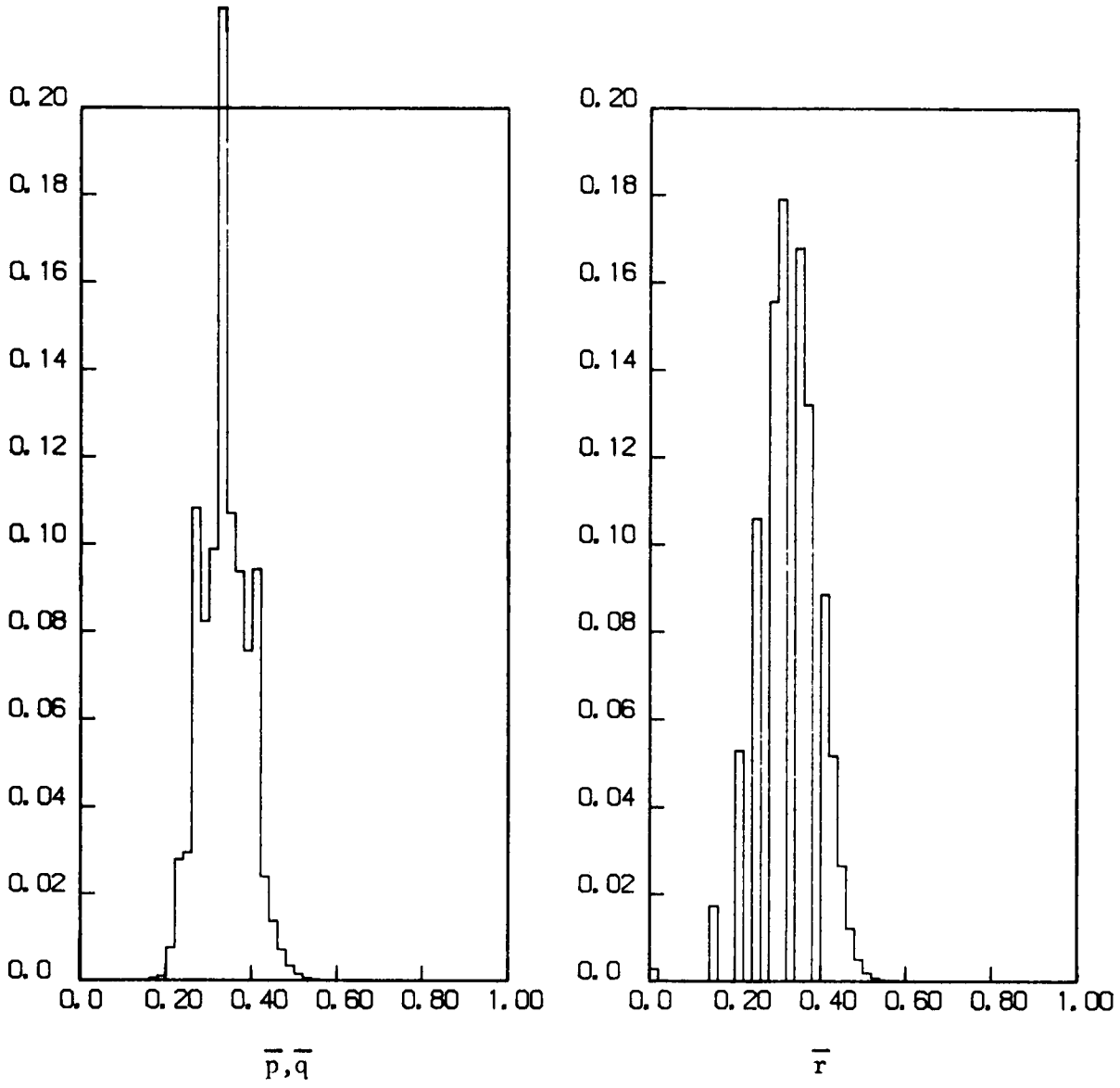


FIGURE 9. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Bernstein's estimator with p, q, and r equal to 1/3 and n equal to 50.

$$\bar{q} = (1 + d)\left(1 - \sqrt{\frac{n_1 + n_2}{n}}\right) \tag{5.29}$$

$$\bar{r} = (1 + d)\left(\sqrt{\frac{n_1}{n}} + d\right) \tag{5.30}$$

The equation for d is given in equation (5.31).

$$d = .5\left(-1 + \sqrt{\frac{n_1 + n_3}{n}} + \sqrt{\frac{n_1 + n_3}{n}} - \sqrt{\frac{n}{n}}\right) \tag{5.31}$$

The formula for d is equal to half of the difference of the sum of the Bernstein's estimators for $\bar{p}$, $\bar{q}$, and $\bar{r}$, from 1.

Table 17 gives the statistics for the improved Bernstein's estimator for $\bar{p}$, $\bar{q}$, and $\bar{r}$ for p, q, and r equal to 1/3 and n varying.

Table 17 shows that the bias and variances of the distributions of p, q, and r decrease as n increases. The $\sqrt{\beta_1}$ and β_2 statistics become closer to normal as n increases. In the rest of the parameter space studied, there are no outstanding abnormalities; the $\sqrt{\beta_1}$ and β_2 statistics remain close to normal.

Figure 10 shows the histograms of the distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for p, q, and r equal to 1/3 and n equal to 50.

As the name implies, there is quite an improvement of these histograms over those for the original Bernstein's estimator. The distributions are unimodal and close to symmetric.

TABLE 17. Statistics for the improved Bernstein's estimator for $\bar{p}$ and $\bar{q}$ (first line) and $\bar{r}$ (second line) for p, q, and r equal to 1/3 and n varying from 30 to 75.

n	μ_1'	σ	σ^2	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3379	.0693	.0048	.0997	.0741	.2993	3.223
	.3227	.0798	.0064	-.1571	322.7	-.3086	3.381
35	.3371	.0638	.0041	.0694	.0526	.2678	3.181
	.3246	.0728	.0053	-.1014	324.6	-.2628	3.370
40	.3365	.0594	.0035	.0509	.0391	.2430	3.148
	.3260	.0674	.0045	-.0673	326.0	-.2203	3.209
45	.3361	.0558	.0031	.0388	.0303	.2234	3.122
	.3269	.0630	.0041	-.0463	326.9	-.1851	3.262
50	.3358	.0528	.0028	.0306	.0241	.2077	3.102
	.3277	.0594	.0035	-.0331	327.7	-.1576	3.206
75	.3349	.0429	.0018	.0126	.0103	.1616	3.055
	.3297	.0479	.0023	-.0102	329.7	-.0928	3.062

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

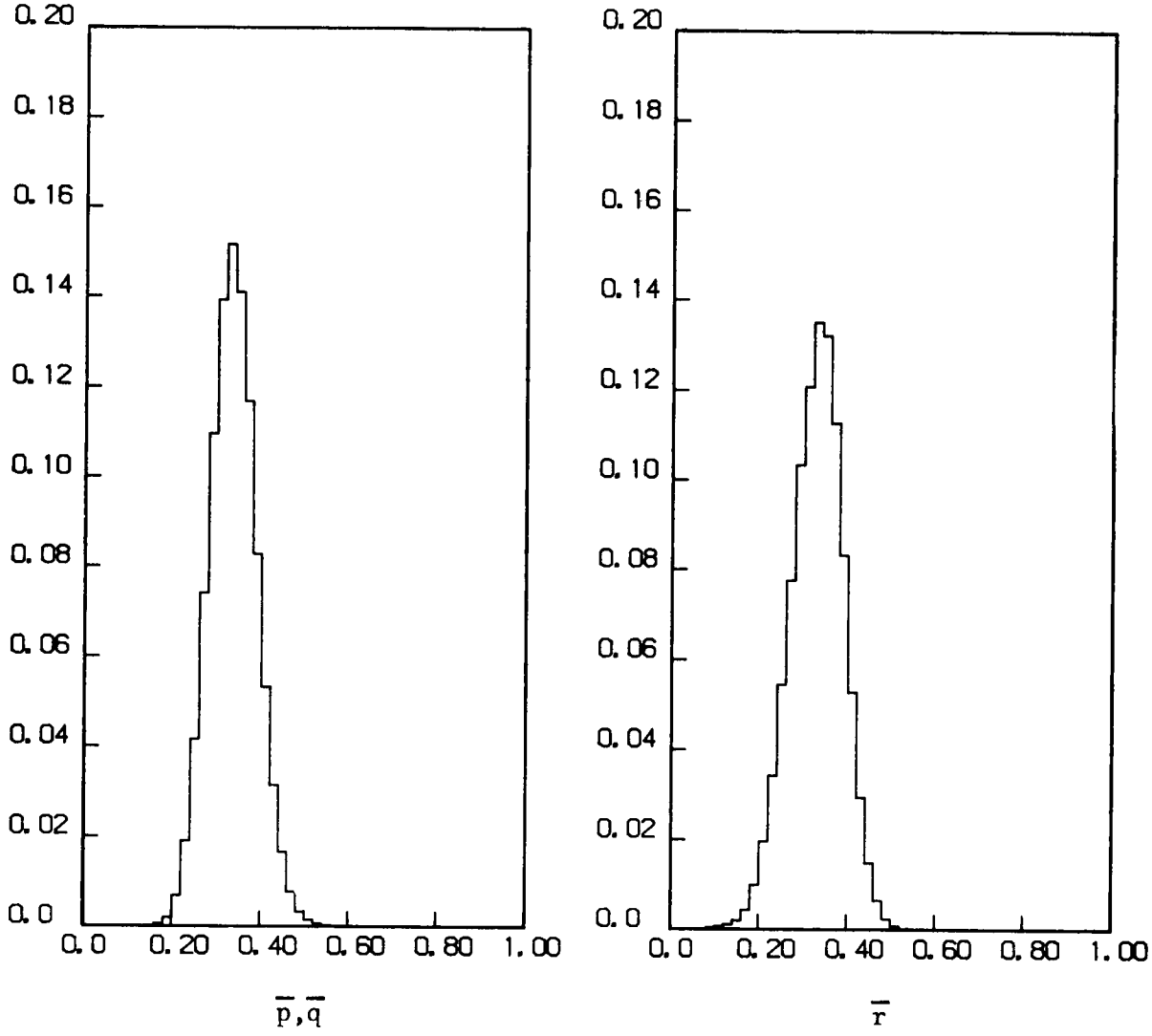


FIGURE 10. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the improved Bernstein's estimator with p, q, and r equal to 1/3 and n equal to 50.

Wiener's estimator

The formulas for Wiener's estimator are given in equations (5.32), (5.33), and (5.34).

$$\bar{p} = \sqrt{\frac{n_1 + n_2}{n}} + \sqrt{\frac{n_1}{n}} \tag{5.32}$$

$$\bar{q} = \sqrt{\frac{n_1 + n_3}{n}} - \sqrt{\frac{n_1}{n}} \tag{5.33}$$

$$\bar{r} = \sqrt{\frac{n_1}{n}} \tag{5.34}$$

Table 18 gives the statistics for Wiener's estimator for p, q, and r equal to 1/3 and n equal to 50.

The bias decreases monotonically toward the value of the parameter as n increases. The variances of the distributions decrease as n increases. The absolute value of $\sqrt{\beta_1}$ and β_2 is getting closer to 3 as n increases. The statistics of the distributions for the rest of the parameter space studied are pretty close to normal.

Figure 11 shows the histograms of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for p, q, and r equal to 1/3 and n equal to 50.

The distribution of $\bar{p}$ and $\bar{q}$ is unimodal and close to symmetric, but the distribution of $\bar{r}$ is not well-behaved. It is identical to the distribution of Bernstein's estimator for $\bar{r}$.

TABLE 18. Statistics for Wiener's estimator for $\bar{p}$, $\bar{q}$ (first line) and $\bar{r}$ (second line) for p, q, and r equal to 1/3 and n varying from 30 to 75.

n	μ_1'	σ	σ^2	μ_3^*	μ_4^*	$\sqrt{\beta_1}$	β_2
30	.3445	.0963	.0093	.5689	.3514	.6373	4.088
	.3185	.0983	.0097	-.7471	318.5	-.7873	4.485
35	.3422	.0870	.0076	.3679	.2309	.5596	4.039
	.3214	.0884	.0078	-.4801	321.4	-.6959	4.491
40	.3406	.0798	.0064	.2452	.1588	.4816	3.905
	.3234	.0809	.0065	-.3166	323.4	-.5986	4.339
45	.3395	.0743	.0055	.1700	.1141	.4138	3.746
	.3248	.0750	.0056	-.2161	324.7	-.5112	4.125
50	.3387	.0698	.0049	.1220	.0853	.3589	3.597
	.3258	.0703	.0050	-.1532	325.8	-.4394	3.940
75	.3367	.0558	.0031	.0387	.0310	.2233	3.202
	.3286	.0560	.0031	-.0460	328.6	-.2617	3.306

*Multiply entries under μ_3 and μ_4 by 10^{-3}.

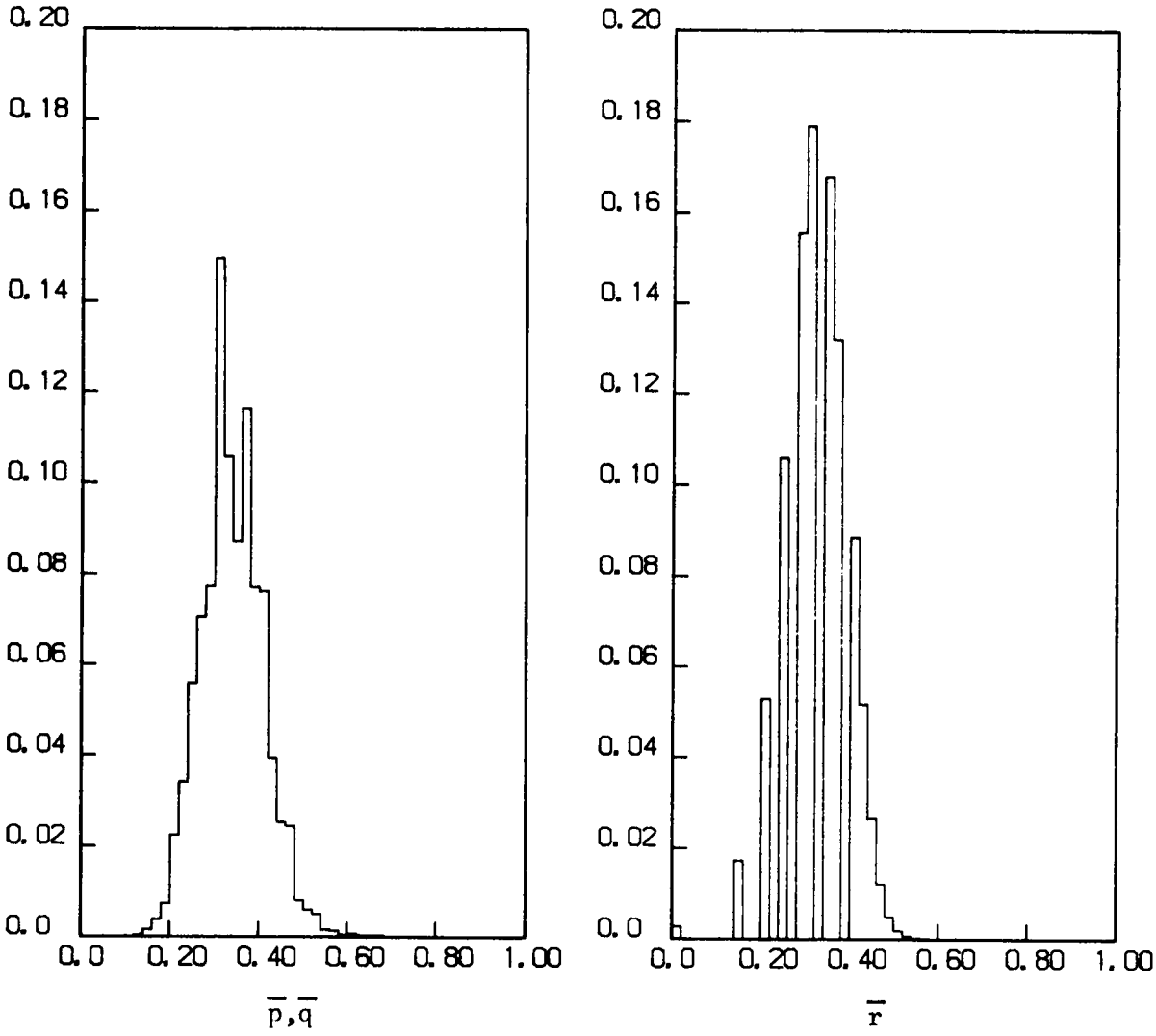


FIGURE 11. Distributions of $\bar{p}$, $\bar{q}$, and $\bar{r}$ for the Wiener's estimator with p, q, and r equal to 1/3 and n equal to 50.

A note on variances

Analytical expressions for the variances of $\bar{p}$, $\bar{q}$, and $\bar{r}$ and the covariance of $\bar{p}$ and $\bar{q}$ have been found for the maximum likelihood, Bernstein's, and Wiener's estimators. They appear in papers by Li [1956, pp. 134-136], Boyd [1956, pp. 24-38], and Chakraborty [1970, pp. 22, 23].

The formulas for the variances of $\bar{p}$, $\bar{q}$, and $\bar{r}$ and the covariance of $\bar{p}$ and $\bar{q}$ for the maximum likelihood estimator are in equations (5.35), (5.36), (5.37), and (5.38).

$$(5.35) \qquad V_{M\bar{p}} = \frac{p(1-p)}{2n} + \frac{p^2}{8n}\left(1 + \frac{r}{pq+r}\right)$$

$$(5.36) \qquad V_{M\bar{q}} = \frac{q(1-q)}{2n} + \frac{q^2}{8n}\left(1 + \frac{r}{pq+r}\right)$$

$$(5.37) \qquad V_{M\bar{r}} = \frac{r(1-r)}{2n} + \frac{(1-r)^2}{8n} + \frac{r(p-q)^2}{8n(pq+r)}$$

$$(5.38) \qquad V_{M\overline{pq}} = -\frac{pq}{2n} + \frac{pq}{8n}\left(1 - \frac{r}{pq+r}\right)$$

Table 19 shows comparison of variances calculated from expressions above those calculated from the exact distributions of the maximum likelihood estimator.

The figures in Table 19 show that the variances calculated by the two methods agree closely except in the case where p is large (.75) and r is small (.15). This is the same part of the parameter space that non-normal values for $\sqrt{\beta_1}$ and β_2 were noticed in the exact distribution of the maximum likelihood estimator (see Table 4, page 32).

TABLE 19. Variances of the maximum likelihood estimator by direct calculation (first line) and by exact distribution (second line) for n equal to 50 and p, q, and r varying.

p q r	$V_{M\bar{p}}$	$V_{M\bar{q}}$	$V_{M\bar{r}}$	$V_{M\bar{pq}}$
.15	.0014	.0014	.0023	-.00022
.15	.0014	.0014	.0023	-.00024
.7				
1/3	.0027	.0027	.0033	-.00104
1/3	.0027	.0027	.0034	-.00104
1/3				
.75	.0042	.0009	.0038	-.00069
.10	.0049	.0011	.0045	-.00082
.15				

The formulas for the variances of $\bar{p}$, $\bar{q}$, and $\bar{r}$ and the covariance of $\bar{p}$ and $\bar{q}$ for Bernstein's estimator are given in equations (5.39), (5.40), (5.41), and (5.42).

$$(5.39) \qquad V_{B\bar{p}} = \frac{p(1 - p)}{2n} + \frac{p^2}{4n}$$

$$(5.40) \qquad V_{B\bar{q}} = \frac{q(1 - q)}{2n} + \frac{q^2}{4n}$$

$$(5.41) \qquad V_{B\bar{r}} = \frac{r(1 - r)}{2r} + \frac{(1 - r)^2}{4n}$$

$$(5.42) \qquad V_{B\bar{pq}} = - \frac{pq}{2n} + \frac{pq\ (1 - r)}{4n\ (pq + r)}$$

Table 20 shows the comparison of variances calculated by exact distribution and by the equations above.

Again there is good agreement between the figures except when p is large (.75) and r is small (.15).

The equations for the variances of $\bar{p}$, $\bar{q}$, and $\bar{r}$ and the covariance of $\bar{p}$ and $\bar{q}$ for Wiener's estimator are given in equations (5.43), (5.44), (5.45), and (5.46).

$$(5.43) \qquad V_{W\bar{p}} = \frac{p(1 - p)}{2n} + \frac{p^2}{4n}\left(\frac{1 + 2q}{p^2 + pr}\right)$$

$$(5.44) \qquad V_{W\bar{q}} = \frac{q(1 - q)}{2n} + \frac{q^2}{4n}\left(1 + \frac{2p}{q^2 + qr}\right)$$

$$(5.45) \qquad V_{W\bar{r}} = \frac{r(1 - r)}{2n} + \frac{(1 - r)^2}{4n}$$

TABLE 20. Variances of Bernstein's estimator by direct calculation (first line) and by exact distribution (second line) for n equal to 50 and p, q, and r varying.

p q r	$V_{B\bar{p}}$	$V_{B\bar{q}}$	$V_{B\bar{r}}$	$V_{B\bar{p}\bar{q}}$
.15	.0014	.0014	.0026	-.00022
.15	.0014	.0014	.0026	-.00022
.7				
1/3	.0028	.0028	.0044	.00097
1/3	.0028	.0028	.0049	-.00099
1/3				
.75	.0047	.0010	.0050	-.00063
.10	.0063	.0010	.0081	-.00069
.15				

$$(5.46) \qquad V_{\overline{Wpq}} = -\frac{pq}{2n} + \frac{pq}{4n}\left(1 + \frac{1}{pq + r}\right)$$

Table 21 gives the variance for Wiener's estimator calculated by exact distribution and by the equations above.

There is quite a large discrepancy between the two methods when p is equal to .75 and r is equal to .15.

Conclusions

In the previous sections, all of the estimators are discussed individually in detail with comments about any unusual or undesirable behavior over the parameter space.

All of the estimators have fairly nice-looking distributions when p, q, and r are equal to 1/3. That is, their statistics are close to normal and the distributions are close to symmetric. Only the distributions of the Bernstein's estimator and Wiener's estimator are not unimodal. All of the estimators but one are evidently consistent, since the expected values approach the parameter values as the sample size increases. The Kullback-Leibler estimator apparently is not consistent, since the means of the distributions do not become closer to the parameter values as n increases.

When the parameter values are not close to 1/3, especially when p or q is large and r is small, several of the estimators have distributions that are far from normal. The maximum likelihood, minimum chi-square, modified chi-square, and Kullback-Leibler estimators all show non-normal values for $\sqrt{\beta_1}$ and β_2 to some extent when p or q is large and r is small.

TABLE 21. Variances of Wiener's estimator by direct calculation (first line) and by exact distribution (second line) for n equal to 50 and p, q, and r varying.

p q r	$V_{W\bar{p}}$	$V_{W\bar{q}}$	$V_{W\bar{r}}$	$V_{W\overline{pq}}$
.15	.0017	.0017	.0026	.00004
.15	.0017	.0017	.0026	.00005
.7				
1/3	.0044	.0044	.0044	.00069
1/3	.0048	.0048	.0049	.00105
1/3				
.75	.0055	.0040	.0049	.00129
.10	.0087	.0064	.0081	.00322
.15				

The bias of any of the estimators does not get out of hand except in the case of the Kullback-Leibler estimator with small r. The maximum likelihood, minimum chi-square, Bernstein's and improved Bernstein's estimators distributions have smaller bias for p, q, and r equal to 1/3 than the other estimators. Of these, only the improved Bernstein's estimator has small bias and close to normal $\sqrt{\beta_1}$ and β_2 over the rest of the parameter space.

Thus there is evidence that for small sample sizes, the improved Bernstein's estimator is as good as any of the general purpose estimators, looking at the whole parameter space. The improved Bernstein's estimator has an added bonus of ease of calculation when compared to other estimators.

Other applications

The concept of examining the exact distributions of estimators appears in Bowman and Shenton [1971, pp. 4-6]. Here Bowman and Shenton examine the exact distribution of the maximum likelihood estimator of the parameter for Fisher's linkage estimation problem.

There are many other probability models in genetics which would lend themselves to this type of study of estimators. Elandt-Johnson [1971, pp. 391-417] discusses the human blood groups. Included are the most important blood group systems, ABO, MNSs, and Rh. Other relatively new blood group systems, such as the Lewis P, Lutheran, and X-linked, are discussed. The maximum likelihood method is derived for the ABO and MNSs systems.

BIBLIOGRAPHY

BIBLIOGRAPHY

Anderson, J. W., 1962, "Least Squares and Best Unbiased Estimates," The Annals of Mathematical Statistics, vol. 33, pp. 266-272.

Bernstein, F., 1925, "Zusammen-fassende Betrachtungen iiber die erblichen Blutstrukturen des menschen," Zschr. induckt. Abstamm., vol. 37.

Bowman, K. O., Hutcheson, K., Odum, E. P., and Shenton, L. R., 1971, "Comments on the Distribution of Indices of Diversity," Statistical Ecology, Many Species, Populations, Ecosystems, and Systems Analysis, vol. 3, pp. 315-366, Pennsylvania State University Press.

Bowman, K. O., and Shenton, L. R., 1971, Note on Fisher's Linkage Estimation Problem, ORNL-4701, Oak Ridge National Laboratory.

Boyd, W. C., 1956, "Variances of Gene Frequency Estimates," The American Journal of Human Genetics, vol. 8, pp. 24-38.

Chakraborty, R., 1970, "Gene Frequency Estimates in the ABO System and Their Efficiencies," Sankya, vol. 32, pp. 21-26.

Elandt-Johnson, R. C., 1971, Probability Models and Statistical Methods in Genetics, John Wiley and Sons.

Fisher, R. A., 1921, "On the Mathematical Foundation of Theoretical Statistics," Philosophical Transactions of the Royal Society, vol. 22, p. 309.

Fisher, R. A., 1925, "Theory of Statistical Information," Proceedings of the Cambridge Philosophical Society, vol. 22, pp. 700-725.

Fisher, R. A., 1956, Statistical Methods and Scientific Inference, Hafner Publishing Company.

Fisher, R. A., 1970, Statistical Methods for Research Workers, Hafner Publishing Company.

Haldane, J. B. S., 1955, "Substitutes for Chi-Square," Biometrika, vol. 42, pp. 265-266.

Hogg, R. V., and Craig, A. T., 1967, Introduction to Mathematical Statistics, The Macmillan Company.

Kempthorne, O., 1969, An Introduction to Genetic Statistics, John Wiley and Sons.

Kendall, M. G., and Stuart, A., 1967, The Advanced Theory of Statistics, Hafner Publishing Company.

Li, C. C., 1956, "The Components of Sampling Variance of ABO Gene Frequency Estimates," The American Journal of Human Genetics, vol. 8, pp. 133-137.

Neyman, J., 1949, "Contribution to the Theory of the Chi-Square Test," Proceedings of the First Berkeley Symposium on Mathematical Statistics and Probability, pp. 239-273.

Rao, C. R., 1961, "Asymptotic Efficiency and Limiting Information," Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 531-545.

Rao, C. R., 1963, "Criteria of Estimation in Large Samples," Sankyā, vol. 25, pp. 189-206.

Rao, C. R., 1968, Linear Statistical Inference and its Applications, John Wiley and Sons.

Sukatme, P. V., 1941, "On Bernstein's Improved Method of Estimating Blood Group Gene Frequencies," Sankya, vol. 6, pp. 85-92.

Westley, G. W., and Watts, J. A., 1970, The Computing Technology Center Numerical Analysis Library, CTC-39, Union Carbide Corporation, Nuclear Division.

Wiener, A. S., Ledederer, M., and Polayes, S. H., 1929, "Studies in Isohemagglutination - I Theoretical Considerations," The Journal of Immunology, vol. 16, pp. 469-482.

APPENDIXES

APPENDIX A

LISTING OF COMPUTER PROGRAM SYSTEM

```
      IMPLICIT REAL*8(A-H,O-Z)                                          MAIN   0
      REAL*8 DMK(6)/3DO,35O,4DO,45O,5DO,75O/                            MAIN   5
      REAL*8 PP(10)/.05O,.15O,.25O,.33O,.4O,.45O,.55O,.65O,             MAIN  10
     > .75O,.85O/                                                       MAIN  15
      REAL*8 QQ(10)/.05O,.15O,.25O,.33O,.4OO,.45O,.25O,.15O,            MAIN  20
     > .1OO,.05O/                                                       MAIN  25
      K = O                                                             MAIN  30
   10 K = K + 1                                                         MAIN  35
      IF (K.GT.3) GO TO 140                                             MAIN  40
      P = 1O/3O                                                         MAIN  45
      Q = 1O/3O                                                         MAIN  50
      R = 1O-P-Q                                                        MAIN  55
      GO TO (20,30,40),K                                                MAIN  60
   20 WRITE(6,10000)                                                    MAIN  65
      GO TO 50                                                          MAIN  70
   30 WRITE(6,10100)                                                    MAIN  75
      GO TO 50                                                          MAIN  80
   40 WRITE(6,10200)                                                    MAIN  85
   50 CONTINUE                                                          MAIN  90
      DO 90 I=1,6                                                       MAIN  95
         GO TO (60,70,80),K                                             MAIN 100
   60    CALL MAX(P,Q,R,DMK(I))                                         MAIN 105
         GO TO 90                                                       MAIN 110
   70    CALL BERN(P,Q,R,DMK(I))                                        MAIN 115
         GO TO 90                                                       MAIN 120
   80    CALL WIENER(P,Q,R,DMK(I))                                      MAIN 125
   90    CONTINUE                                                       MAIN 130
      DO 130 I=1,10                                                     MAIN 135
         P = PP(I)                                                      MAIN 140
         IF (P.EQ..33D0) P = 1D0/3D0                                    MAIN 145
         Q = QQ(I)                                                      MAIN 150
         IF (Q.EQ..33D0) Q = 1D0/3D0                                    MAIN 155
         R = 1D0-P-Q                                                    MAIN 160
         GO TO (100,110,120),K                                          MAIN 165
  100    CALL MAX(P,Q,R,50D0)                                           MAIN 170
         GO TO 130                                                      MAIN 175
  110    CALL BERN(P,Q,R,50D0)                                          MAIN 180
         GO TO 130                                                      MAIN 185
  120    CALL WIENER(P,Q,R,50D0)                                        MAIN 190
  130    CONTINUE                                                       MAIN 195
      GO TO 10                                                          MAIN 200
  140 RETURN                                                            MAIN 205
10000 FORMAT ('1MAXIMUM LIKELIHOOD MATRIX OF COVARIANCES')              MAIN 210
10100 FORMAT ('1  BERNSTEIN''S METHOD MATRIX OF COVARIANCES')           MAIN 215
10200 FORMAT ('1WIENER''S METHOD   MATRIX OF COVARIANCES')              MAIN 220
      END                                                               MAIN 225
```

```
      SUBROUTINE WIENER(P,Q,R,DMK)                                          WIEN   0
      IMPLICIT REAL*8(A-H,O-Z)                                              WIEN   5
      C = .5D0/DMK                                                          WIEN  10
      D = .25D0/DMK                                                         WIEN  15
      VP = C*P*(1D0-P) + D*P*P*(1D0+2D0*Q/(P*P+P*R))                        WIEN  20
      VQ = C*Q*(1D0-Q) + D*Q*Q*(1D0 + 2D0*P/(Q*Q+Q*R))                      WIEN  25
      VPQ = -C*P*Q + D*P*Q*(1D0 + 1D0/(P*Q+R))                              WIEN  30
      VR = R*(1D0-R)*C + (1D0-R)**2*D                                       WIEN  35
      WRITE(6,10000) P,Q,R,DMK                                              WIEN  40
      WRITE(6,10100) VP,VQ,VR,VPQ                                           WIEN  45
      RETURN                                                                WIEN  50
10000 FORMAT (' '/' P=',F6.3,2X,'Q=',F6.3,2X,'R=',F6.3,2X,'N=',F4.0)        WIEN  55
10100 FORMAT ('0',4D16.8)                                                   WIEN  60
      END                                                                   WIEN  65
```

```
      SUBROUTINE BERN(P,Q,R,DMK)                                        BERN   0
      IMPLICIT REAL*8(A-H,O-Z)                                          BERN   5
      T = 1DO-R/(P*Q+R)                                                 BERN  10
      C = .5DO/DMK                                                      BERN  15
      D = .25DO/DMK                                                     BERN  20
      VP = C*P*(1DO-P) + D*P*P                                          BERN  25
      VQ = C*Q*(1DO-Q) + D*Q*Q                                          BERN  30
      VPQ = -C*P*Q + D*P*Q*T                                            BERN  35
      VR = R*(1DO-R)*C + (1DO-R)**2*D                                   BERN  40
      WRITE(6,10000) P,Q,R,DMK                                          BERN  45
      WRITE(6,10100) VP,VQ,VR,VPQ                                       BERN  50
      RETURN                                                            BERN  55
10000 FORMAT (' '/' P=',F6.3,2X,'Q=',F6.3,2X,'R=',F6.3,2X,'N=',F4.0)    BERN  60
10100 FORMAT ('0',4D16.8)                                               BERN  65
      END                                                               BERN  70
```

```
      SUBROUTINE MAX(P,Q,R,DMK)                                          MAX    0
      IMPLICIT REAL*8(A-H,O-Z)                                           MAX    5
      C = .5DO/DMK                                                       MAX   10
      D = .125DO/DMK                                                     MAX   15
      T = R/(P*Q+R)                                                      MAX   20
      VP = C*P*(1DO-P) + D*P*P*(1DO+T)                                   MAX   25
      VQ = C*Q*(1DO-Q) + D*Q*Q*(1DO+T)                                   MAX   30
      VPQ = -C*P*Q + D*P*Q*(1DO-T)                                       MAX   35
      VR = R*(1DO-R)*C + (1DO-R)**2*D + (P-Q)**2*T*D                     MAX   40
      WRITE(6,10000) P,Q,R,DMK                                           MAX   45
      WRITE(6,10100) VP,VQ,VR,VPQ                                        MAX   50
      RETURN                                                             MAX   55
10000 FORMAT (' '/' P=',F6.3,2X,'Q=',F6.3,2X,'R=',F6.3,2X,'N=',F4.0)     MAX   60
10100 FORMAT ('0',4D16.8)                                                MAX   65
      END                                                                MAX   70
```

```
      IMPLICIT REAL*8(A-H,O-Z)                                          MAIN    0
      DIMENSION ALPHA(10),P(10),Q(10),R(10),SUM(10)                     MAIN    5
      DIMENSION CM10(10),CM01(10),CM001(10),CM20(10),CM02(10),CM002(10) MAIN   10
     > ,CM30(10),CM03(10),CM003(10),CM40(10),CM04(10),CM004(10),        MAIN   15
     > CM12(10),CM13(10),CM23(10)                                       MAIN   20
      DIMENSION RHO12(10),RHO13(10),RHO23(10),B1R10(10),B1R01(10),      MAIN   25
     > B1R001(10),B21(10),B22(10),B23(10),RCM20(10),RCM02(10),RCM002(10)MAIN   30
      DIMENSION VEC(32),FACLOG(201),PRO(32,10),PROL(32,10)              MAIN   35
      DIMENSION CE(50),CEL(50,10),CP(50),CEQ(50,10),CQ(50),CER(50,10),  MAIN   40
     > CR(50)                                                           MAIN   45
      DIMENSION ICELQ(7), ICELP(7),ICELR(7)                             MAIN   50
      COMMON VEC,DMK                                                    MAIN   55
      COMMON/RHAT/REST
      COMMON/LOGFAC/FACLOG                                              MAIN   60
      COMMON /FAIL/ NFL                                                 MAIN   65
      COMMON/ICASE/IC                                                   MAIN   70
      DIMENSION PS(10)                                                  MAIN   75
      CALL ERRSET(209,256,0,0)                                          MAIN   80
      CALL ERRSET(207,256,0,0)                                          MAIN   85
      CALL ERRSET(261,256,0,0)                                          MAIN   90
      FACLOG(1) = 0.0D0                                                 MAIN   95
      FACLOG(2) = 0.0D0                                                 MAIN  100
      DO 10 I = 3,201                                                   MAIN  105
         DUM = I-1                                                      MAIN  110
   10    FACLOG(I) = FACLOG(I-1) + DLOG(DUM)                            MAIN  115
   20 CONTINUE                                                          MAIN  120
      READ(5,10000,END=230) IC,ALPHA                                    MAIN  125
      READ(5,10100) (P(I),Q(I),I=1,IC)                                  MAIN  130
      DO 50 J=1,IC                                                      MAIN  135
         IF (P(J).NE..33D0) GO TO 30                                    MAIN  140
         P(J) = 1D0/3D0                                                 MAIN  145
         Q(J) = 1D0/3D0                                                 MAIN  150
   30    CONTINUE                                                       MAIN  155
         R(J) = 1D0-P(J)-Q(J)                                           MAIN  160
         PRO(1,J) = R(J)*R(J)                                           MAIN  165
         PRO(2,J) = P(J)*P(J) + 2D0*P(J)*R(J)                           MAIN  170
         PRO(3,J) = Q(J)*Q(J) + 2D0*Q(J)*R(J)                           MAIN  175
         PRO(4,J) = 2D0*P(J)*Q(J)                                       MAIN  180
         JC = 4                                                         MAIN  185
         DO 40 I = 1,JC                                                 MAIN  190
            DUM = PRO(I,J)                                              MAIN  195
   40       PROL(I,J) = DLOG(DUM)                                       MAIN  200
   50    CONTINUE                                                       MAIN  205
      NFL = 0                                                           MAIN  210
      READ(5,10200) MK ,IPUNCH                                          MAIN  215
C     MK = SAMPLE SIZE                                                  MAIN  220
      DO 60 J=1,IC                                                      MAIN  225
         CM10(J)= 0D0                                                   MAIN  230
         CM01(J)= 0D0                                                   MAIN  235
         CM12(J)= 0D0                                                   MAIN  240
         CM13(J)= 0D0                                                   MAIN  245
         CM23(J)= 0D0                                                   MAIN  250
         CM20(J)= 0D0                                                   MAIN  255
         CM02(J)= 0D0                                                   MAIN  260
         CM30(J)= 0D0                                                   MAIN  265
         CM03(J)= 0D0                                                   MAIN  270
         CM40(J)= 0D0                                                   MAIN  275
         CM04(J)= 0D0                                                   MAIN  280
         CM001(J) = 0D0                                                 MAIN  285
```

```
      CM002(J) = 0D0                                                    MAIN 290
      CM003(J) = 0D0                                                    MAIN 295
      CM004(J) = 0D0                                                    MAIN 300
      SUM(J) = 0D0                                                      MAIN 305
 60   CONTINUE                                                          MAIN 310
   DMK = MK                                                             MAIN 315
   CALL CELL(CE,CEL,CEQ,CER)                                            MAIN 320
   DO 70 I = 1,7                                                        MAIN 325
      ICELR(I) = 0                                                      MAIN 330
      ICELQ(I) =0                                                       MAIN 335
 70   ICELP(I) = 0                                                      MAIN 340
   IF (MK.GE.JC) GO TO 80                                               MAIN 345
   JCC = MK                                                             MAIN 350
   GO TO 90                                                             MAIN 355
 80 JCC = JC                                                            MAIN 360
 90 CONTINUE                                                            MAIN 365
   IX2 = 0                                                              MAIN 370
   IX1 = 1                                                              MAIN 375
   IID = 1                                                              MAIN 380
   DO 100 I=1,3                                                         MAIN 385
100   VEC(I) = 0D0                                                      MAIN 390
   VEC(4) = MK                                                          MAIN 395
   CALL COMB(VEC,MK,&170)                                               MAIN 400
110 CONTINUE                                                            MAIN 405
   IF (IX2.LT.IID*300000) GO TO 120                                     MAIN 410
   IID = IID + 1                                                        MAIN 415
   WRITE(6,10300) (VEC(I),I=1,4)                                        MAIN 420
120 CONTINUE                                                            MAIN 425
   IX1 = IX1 + 1                                                        MAIN 430
   DO 130 J=1,IC                                                        MAIN 435
130   PS(J) = FACLOG(MK+1)                                              MAIN 440
   DO 140 JK = 1,JC                                                     MAIN 445
      IF (VEC(JK).LE.0D0) GO TO 140                                     MAIN 450
      IDUM = VEC(JK) + 1                                                MAIN 455
      DO 140 J=1,IC                                                     MAIN 460
         PS(J) = PS(J) - FACLOG(IDUM)+VEC(JK)*PROL(JK,J)                MAIN 465
140      CONTINUE                                                       MAIN 470
   REST = 100D0
   CALL MAX2G(PEST,QEST)
   IF (REST .EQ. 100D0) REST = 1D0-PEST-QEST
   CALL CELLI(ICELP,ICELQ,ICELR,PEST,QEST,REST)                         MAIN 485
   DO 150 J=1,IC                                                        MAIN 490
      IF (DABS(PS(J)).GE.174.6D0) GO TO 150                             MAIN 495
      PS(J) = DEXP(PS(J))                                               MAIN 500
      PP = PS(J)                                                        MAIN 505
      CM10(J) = CM10(J) + PP*PEST                                       MAIN 510
      CM01(J) = CM01(J) + PP*QEST                                       MAIN 515
      CM12(J) = CM12(J) + PP*PEST*QEST                                  MAIN 520
      CM20(J) = CM20(J) + PP*PEST*PEST                                  MAIN 525
      CM02(J) = CM02(J) + PP*QEST*QEST                                  MAIN 530
      CM03(J) = CM03(J) + PP*QEST**3                                    MAIN 535
      CM30(J) = CM30(J) + PP*PEST**3                                    MAIN 540
      CM04(J) = CM04(J) + PP*QEST**4                                    MAIN 545
      CM40(J) = CM40(J) + PP*PEST**4                                    MAIN 550
      CM001(J) = CM001(J) + PP*REST                                     MAIN 555
      CM002(J) = CM002(J) + PP*REST*REST                                MAIN 560
      CM003(J) = CM003(J) + PP*REST**3                                  MAIN 565
      CM004(J) = CM004(J) + PP*REST**4                                  MAIN 570
      CM13(J) = CM13(J) + PP*PEST*REST                                  MAIN 575
```

```
      CM23(J) = CM23(J) + PP*QEST*REST                                  MAIN 580
      SUM(J) = SUM(J) + PP                                              MAIN 585
150   CONTINUE                                                          MAIN 590
      CALL CELL2(PEST,QEST,REST,PS,CE,CEL,CEQ,CER)                      MAIN 595
      IX2 = IX2 + 1                                                     MAIN 600
160 CONTINUE                                                            MAIN 605
      CALL GO (&170)                                                    MAIN 61
      GO TO 110                                                         MAIN 615
170 CONTINUE                                                            MAIN 620
      DO 210 J=1,IC                                                     MAIN 625
        CM40 (J) = CM40 (J)-4D0*CM30 (J)*CM10 (J)+6D0*CM20 (J)*CM10 (J)MAIN 630
     >    **2 - 3D0*CM10 (J)**4                                        MAIN 635
        CM04 (J) = CM04 (J)-4D0*CM03 (J)*CM01 (J)+6D0*CM02 (J)*CM01 (J)MAIN 640
     >    **2 - 3D0*CM01 (J)**4                                        MAIN 645
        CM004(J) = CM004(J)-4D0*CM003(J)*CM001(J)+6D0*CM002(J)*CM001(J)MAIN 650
     >    **2 - 3D0*CM001(J)**4                                        MAIN 655
        CM30 (J) = CM30 (J)-3D0*CM20 (J)*CM10 (J) + 2D0*CM10 (J)**3   MAIN 660
        CM03 (J) = CM03 (J)-3D0*CM02 (J)*CM01 (J) + 2D0*CM01 (J)**3   MAIN 665
        CM003(J) = CM003(J)-3D0*CM002(J)*CM001(J) + 2D0*CM001(J)**3   MAIN 670
        CM12(J) = CM12(J) - CM10(J)*CM01(J)                           MAIN 675
        CM13(J) = CM13(J) - CM10(J)*CM001 (J)                         MAIN 680
        CM23(J) = CM23(J) - CM01(J)*CM001 (J)                         MAIN 685
        CM20 (J) = CM20 (J) - CM10 (J)**2                             MAIN 690
        CM02 (J) = CM02 (J) - CM01 (J)**2                             MAIN 695
        CM002(J) = CM002(J) - CM001(J)**2                             MAIN 700
        RHO12(J) = CM12(J)/DSQRT(CM20(J)*CM02(J))                     MAIN 705
        RHO13(J) = CM13(J)/DSQRT(CM20(J)*CM002(J))                    MAIN 710
        RHO23(J) = CM23(J)/DSQRT(CM02(J)*CM002(J))                    MAIN 715
        B1R10(J) = CM30(J)/CM20(J)**1.5D0                             MAIN 720
        B1R01(J) = CM03(J)/CM02(J)**1.5D0                             MAIN 725
        B1R001(J) = CM003(J)/CM002(J)**1.5D0                          MAIN 730
        B21(J) = CM40(J)/CM20(J)**2                                   MAIN 735
        B22(J) = CM04(J)/CM02(J)**2                                   MAIN 740
        B23(J) = CM004(J)/CM002(J)**2                                 MAIN 745
        RCM20(J) = DSQRT(CM20(J))                                     MAIN 750
        RCM02(J) = DSQRT(CM02(J))                                     MAIN 755
        RCM002(J) = DSQRT(CM002(J))                                   MAIN 760
        WRITE(6,10400) ALPHA,P(J),Q(J),R(J),(PRO(I,J),I=1,4),MK,SUM(J) MAIN 765
        WRITE(6,10500) CM10(J),RCM20(J),CM20(J),CM30(J),CM40(J),      MAIN 770
     >    B1R10(J), B21(J)                                            MAIN 775
        WRITE(6,10600) CM01(J),RCM02(J),CM02(J),CM03(J),CM04(J),      MAIN 780
     >    B1R01(J), B22(J)                                            MAIN 785
        WRITE(6,10700) CM001(J),RCM002(J),CM002(J),CM003(J),CM001(J), MAIN 790
     >    B1R001(J),B23(J)                                            MAIN 795
        WRITE(6,10800) CM12(J),RHO12(J),CM13(J),RHO13(J),CM23(J),     MAIN 800
     >    RHO23(J)                                                    MAIN 805
        WRITE(6,10900)(CEL(I,J),I=1,50)                               MAIN 810
        CP(1) = CEL(1,J)                                              MAIN 815
        DO 180 I = 2,50                                               MAIN 820
180       CP(I) = CP(I-1) + CEL(I,J)                                  MAIN 825
        WRITE(6,10900) CP                                             MAIN 830
        WRITE(6,10900)(CEQ(I,J),I=1,50)                               MAIN 835
        CQ(1) = CEQ(1,J)                                              MAIN 840
        DO 190 I=2,50                                                 MAIN 845
190       CQ(I) = CQ(I-1) + CEQ(I,J)                                  MAIN 850
        WRITE(6,10900) CQ                                             MAIN 855
        WRITE(6,10900)(CER(I,J),I=1,50)                               MAIN 860
        CR(1) = CER(1,J)                                              MAIN 865
        DO 200 I=2,50                                                 MAIN 870
```

```
  200       CR(I) = CR(I-1) + CER(I,J)                                    MAIN 875
            WRITE(6,10900) CR                                             MAIN 880
            WRITE(6,11000) ICELP,ICELQ,ICELR                              MAIN 885
            IF (IPUNCH.GT.0) WRITE(7,11100) ALPHA,P(J),Q(J),R(J)          MAIN 890
            IF (IPUNCH.GT.0) WRITE(7,11200)(CEL(I,J),I=1,50)              MAIN 895
            IF (IPUNCH.GT.0) WRITE(7,11200)(CEQ(I,J),I=1,50)              MAIN 900
            IF (IPUNCH.GT.0) WRITE(7,11200)(CER(I,J),I=1,50)              MAIN 905
  210       CONTINUE                                                      MAIN 910
        WRITE(6,11300) NFL                                                MAIN 915
  220   CONTINUE                                                          MAIN 920
        GO TO 20                                                          MAIN 925
  230   RETURN                                                            MAIN 930
10000   FORMAT (I4,10A4)                                                  MAIN 935
10100   FORMAT (2D10.0)                                                   MAIN 940
10200   FORMAT (2I3)                                                      MAIN 945
10300   FORMAT(1HO,32I3)                                                  MAIN 950
10400   FORMAT ('1',10A4,5X,'P=',F5.3,2X,'Q=',F5.3,2X,'R=',F5.3,/ ' PRO=',MAIN 955
     > 4F15.8/' MK=',I5,2X,'SUM=',F15.8)                                  MAIN 960
10500   FORMAT (' STATISTICS FOR P',7F15.8)                               MAIN 965
10600   FORMAT (' STATISTICS FOR Q', 7F15.8)                              MAIN 970
10700   FORMAT (' STATISTICS FOR R',7F15.8)                               MAIN 975
10800   FORMAT(' COVARIANCE PQ =',D15.8,'CORRELATION COEFF PQ =',D15.8/   MAIN 980
     > ' COVARIANCE PR =',D15.8,'CORRELATION COEFF PR =',D15.8/            MAIN 985
     > ' COVARIANCE QR =',D15.8,'CORRELATION COEFF QR =',D15.8)            MAIN 990
10900   FORMAT(1HO,8F15.8/(1X,8F15.8))                                    MAIN 995
11000   FORMAT ('0',7I15/' ',7I15/' ',7I15)                               MAIN1000
11100   FORMAT (10A4,3F10.3)                                              MAIN1005
11200   FORMAT (5F15.8)                                                   MAIN1010
11300   FORMAT ('1NUMBER OF CASES THAT DID NOT CONVERGE IS',I10)          MAIN1015
        END                                                               MAIN1020
        SUBROUTINE CELL( CE,CEL,CEQ,CER)                                  CELL   0
        IMPLICIT REAL*8(A-H,O-Z)                                          CELL   5
        DIMENSION CE(1),CEL(50,1),CER(50,1),CEQ(50,1)                     CELL  10
        COMMON /ICASE/ IC                                                 CELL  15
        DO 10 J=1,IC                                                      CELL  20
           DO 10 I = 1,50                                                 CELL  25
              CEL(I,J) = 0D0                                              CELL  30
              CEQ(I,J) = 0D0                                              CELL  35
              CER(I,J) = 0D0                                              CELL  40
  10          CONTINUE                                                    CELL  45
        DNL = 1.0D0/50D0                                                  CELL  50
        CE(1) = DNL                                                       CELL  55
        DO 20 I = 2,50                                                    CELL  60
  20       CE(I) = CE(I-1) + DNL                                          CELL  65
        RETURN                                                            CELL  70
        END                                                               CELL  75
        SUBROUTINE CELL2(PEST,QEST,REST,PP,CE,CEL,CEQ,CER)                CELL   0
        IMPLICIT REAL*8(A-H,O-Z)                                          CELL   5
        COMMON /ICASE/IC                                                  CELL  10
        DIMENSION CE(1),CEL(50,10),CEQ(50,10),CER(50,10),PP(10)           CELL  15
        DO 10 I = 1,50                                                    CELL  20
           K = I                                                          CELL  25
           IF (PEST.LT.CE(I)) GO TO 20                                    CELL  30
  10       CONTINUE                                                      CELL  35
  20    DO 30 J=1,IC                                                      CELL  40
           IF (DABS(PP(J)).GT.174.6D0) GO TO 30                           CELL  45
           CEL(K,J) = CEL(K,J) + PP(J)                                    CELL  50
  30       CONTINUE                                                      CELL  55
        DO 40 I=1,50                                                      CELL  60
```

```
      K = I                                                              CELL  65
      IF (QEST.LT.CE(I)) GO TO 50                                        CELL  70
40    CONTINUE                                                           CELL  75
50 DO 60 J=1,IC                                                          CELL  80
      IF (DABS(PP(J)).GT.174.6D0) GO TO 60                               CELL  85
      CEQ(K,J) = CEQ(K,J) + PP(J)                                        CELL  90
60    CONTINUE                                                           CELL  95
   DO 70 I=1,50                                                          CELL 100
      K = I                                                              CELL 105
      IF (REST.LT.CE(I)) GO TO 80                                        CELL 110
70    CONTINUE                                                           CELL 115
80 DO 90 J=1,IC                                                          CELL 120
      IF (DABS(PP(J)).GT.174.6D0) GO TO 90                               CELL 125
      CER(K,J) = CER(K,J) + PP(J)                                        CELL 130
90    CONTINUE                                                           CELL 135
   RETURN                                                                CELL 140
   END                                                                   CELL 145
   SUBROUTINE CELLI(ICELP,ICELQ,ICELR,PEST,QEST,REST)                    CELL   0
   IMPLICIT REAL*8(A-H,O-Z)                                              CELL   5
   DIMENSION ICELP(1),ICELQ(1),ICELR(1),DUM(7)                           CELL  10
   DUM(1) = 0.002                                                        CELL  15
   DUM(2) = 0.2                                                          CELL  20
   DUM(3) = 0.4                                                          CELL  25
   DUM(4) = 0.6                                                          CELL  30
   DUM(5) = 0.8                                                          CELL  35
   DUM(6) = 0.998                                                        CELL  40
   DUM(7) = 1.0D0                                                        CELL  45
   DO 10 I = 1,7                                                         CELL  50
      II = I                                                             CELL  55
      IF (PEST.LT.DUM(I)) GO TO 20                                       CELL  60
10    CONTINUE                                                           CELL  65
20 ICELP(II) = ICELP(II) + 1                                             CELL  70
   DO 30 I=1,7                                                           CELL  75
      II = I                                                             CELL  80
      IF (QEST.LT.DUM(I)) GO TO 40                                       CELL  85
30    CONTINUE                                                           CELL  90
40 ICELQ(II) = ICELQ(II) + 1                                             CELL  95
   DO 50 I=1,7                                                           CELL 100
      II = I                                                             CELL 105
      IF (REST.LT.DUM(I)) GO TO 60                                       CELL 110
50    CONTINUE                                                           CELL 115
60 ICELR(II) = ICELR(II) + 1                                             CELL 120
   RETURN                                                                CELL 125
   END                                                                   CELL 130
   SUBROUTINE COMB(A,N,*)                                                COMB   0
   REAL*8A(4),D                                                          COMB   5
   D = N                                                                 COMB  10
   L=N                                                                   COMB  15
   K=N                                                                   COMB  20
   RETURN                                                                COMB  25
   ENTRY GO (*)                                                          COMB  30
10 IF (A(3).EQ.K) GO TO 20                                               COMB  35
   A(3) = A(3) + 1D0                                                     COMB  40
   A(4) = A(4) -1D0                                                      COMB  45
   RETURN                                                                COMB  50
20 IF (A(2).EQ.L) GO TO 30                                               COMB  55
   A(2) = A(2) + 1D0                                                     COMB  60
   A(3) = 0D0                                                            COMB  65
   A(4) = D -A(2) -A(1)                                                  COMB  70
```

```
      K = A(4)                                                          COMB   75
      RETURN                                                            COMB   80
   30 IF (L.EQ.0) RETURN 1                                              COMB   85
      A(1) = A(1) + 1D0                                                 COMB   90
      A(2) = 0D0                                                        COMB   95
      A(3) = 0D0                                                        COMB  100
      A(4) = D -A(1)                                                    COMB  105
      K = A(4)                                                          COMB  110
      L = L-1                                                           COMB  115
      RETURN                                                            COMB  120
      END                                                               COMB  125
      SUBROUTINE CONSTR                                                 CONS    0
      REAL*8 B,G                                                        CONS    5
      COMMON/BAMRWH/G(20,50),B(50)                                      CONS   10
      B(1) = 1D0                                                        CONS   15
      B(2) = 1D0                                                        CONS   20
      B(3) = 0D0                                                        CONS   25
      B(4) = 0D0                                                        CONS   30
      B(5) = 1D0                                                        CONS   35
      B(6) = 0D0                                                        CONS   40
      G(1,1) = 1D0                                                      CONS   45
      G(2,1) = 0D0                                                      CONS   50
      G(1,2) = 0D0                                                      CONS   55
      G(2,2) = 1D0                                                      CONS   60
      G(1,3) = -1D0                                                     CONS   65
      G(2,3) = 0D0                                                      CONS   70
      G(1,4) = 0D0                                                      CONS   75
      G(2,4) = -1D0                                                     CONS   80
      G(1,5) = 1D0                                                      CONS   85
      G(2,5) = 1D0                                                      CONS   90
      G(1,6) = -1D0                                                     CONS   95
      G(2,6) = -1D0                                                     CONS  100
      RETURN                                                            CONS  105
      END                                                               CONS  110
```

```
      SUBROUTINE LESQ(P,Q)                                                    LESQ   0
      IMPLICIT REAL*8(A-H,O-Z)                                                LESQ   5
      COMMON /FAIL/ NFL                                                       LESQ  10
      COMMON VEC(32),DMK                                                      LESQ  15
      EXTERNAL CONSTR,LESVAL,LESGR                                            LESQ  20
      DIMENSION XB(20)                                                        LESQ  25
      REAL*8 EPS(20)/2*1D-3/                                                  LESQ  30
      XB(1) = 1DO-DSQRT((VEC(1)+VEC(3))/DMK)                                  LESQ  35
      XB(2) = 1DO-DSQRT((VEC(1)+VEC(2))/DMK)                                  LESQ  40
      IW = 0                                                                  LESQ  45
      FLR = ODO                                                               LESQ  50
      IF (XB(1)+XB(2).LE.1DO) GO TO 10                                        LESQ  55
      XB(1) = .5DO                                                            LESQ  60
      XB(2) = .5DO                                                            LESQ  65
   10 CALL KEELE(2,6,0,0,5,20,IW,XB,FB,1D-5,1D-7,1D-7,1D-7,1D-7,LESVAL,       LESQ  70
     > LESGR ,CONSTR,IFL,FLR,1D-3,EPS)                                        LESQ  75
      IF (IFL.GT.0) NFL = NFL + 1                                             LESQ  80
      IF (IW.GT.0) WRITE(6,10000) (VEC(I),I=1,4)                              LESQ  85
      IF (IFL.GT.0.AND.IW.GT.0) WRITE(6,10100) IFL                            LESQ  90
      P = XB(1)                                                               LESQ  95
      Q = XB(2)                                                               LESQ 100
      RETURN                                                                  LESQ 105
10000 FORMAT ('0',4F10.5)                                                     LESQ 110
10100 FORMAT ('0IFL=',I5)                                                     LESQ 115
      END                                                                     LESQ 120
```

```
      SUBROUTINE LESVAL(X,F)                                                LESV    0
      IMPLICIT REAL*8(A-H,O-Z)                                              LESV    5
      COMMON VEC(32),DMK                                                    LESV   10
      DIMENSION X(1)                                                        LESV   15
      P = X(1)                                                              LESV   20
      Q = X(2)                                                              LESV   25
      T1 = 1D0-P-Q                                                          LESV   30
      T1S = T1*T1                                                           LESV   35
      T2 = -P*P+2D0*P-2D0*P*Q                                               LESV   40
      T3 = -Q*Q+2D0*Q-2D0*P*Q                                               LESV   45
      T4 = 2D0*P*Q                                                          LESV   50
      F=(VEC(1)-DMK*T1S)**2 + (VEC(2)-DMK*T2)**2 + (VEC(3)-DMK*T3)**2 +     LESV   55
     > (VEC(4)-DMK*T4)**2                                                   LESV   60
      RETURN                                                                LESV   65
      END                                                                   LESV   70
```

```
      SUBROUTINE LESGR (X,F)                                                    LESG   0
      IMPLICIT REAL*8(A-H,O-Z)                                                  LESG   5
      COMMON VEC(32),DMK                                                        LESG  10
      DIMENSION X(1),F(1)                                                       LESG  15
      P = X(1)                                                                  LESG  20
      Q = X(2)                                                                  LESG  25
      T1 = 1D0-P-Q                                                              LESG  30
      T1S = T1*T1                                                               LESG  35
      T2 = -P*P+2D0*P-2D0*P*Q                                                   LESG  40
      T3 = -Q*Q+2D0*Q-2D0*P*Q                                                   LESG  45
      T4 = 2D0*P*Q                                                              LESG  50
      U1 = 4D0*(VEC(1)-DMK*T1S)*DMK*T1                                          LESG  55
      U2 = (VEC(2)-DMK*T2)                                                      LESG  60
      U3 = (VEC(3)-DMK*T3)                                                      LESG  65
      U4 = (VEC(4)-DMK*T4)                                                      LESG  70
      F(1) = U1 - 4D0*U2*DMK*T1 + 4D0*U3*DMK*Q - 4D0*U4*DMK*Q                   LESG  75
      F(2) = U1 + 4D0*U2*DMK*P - 4D0*U3*DMK*T1 - 4D0*U4*DMK*P                   LESG  80
      RETURN                                                                    LESG  85
      END                                                                       LESG  90
```

```
      SUBROUTINE HD2(P,Q)                                                    HD2    0
      IMPLICIT REAL*8(A-H,O-Z)                                               HD2    5
      COMMON VEC(32) ,DMK                                                    HD2   10
      COMMON /FAIL/ NFL                                                      HD2   15
      EXTERNAL HDVAL,HDGRAD,CONSTR                                           HD2   20
      DIMENSION XB(20)                                                       HD2   25
      REAL*8 EPS(20)/2*1D-3/                                                 HD2   30
      XB(1) = 1DO-DSQRT((VEC(1)+VEC(3))/DMK)                                 HD2   35
      XB(2) = 1DO-DSQRT((VEC(1)+VEC(2))/DMK)                                 HD2   40
      FLR = 0DO                                                              HD2   45
      IW = 0                                                                 HD2   50
      IF (XB(1)+XB(2).LE.1DO) GO TO 10                                       HD2   55
      XB(1) = .5DO                                                           HD2   60
      XB(2) = .5DO                                                           HD2   65
   10 CALL KEELE(2,6,0,0,5,25,IW,XB,F8,1D-5,1D-7,1D-7,1D-7,1D-7,HDVAL,       HD2   70
     > HDGRAD,CONSTR,IFL,FLR,1D-3,EPS)                                       HD2   75
      IF (IFL.GT.0) NFL = NFL + 1                                            HD2   80
      IF (IFL.GT.0.AND.IW.GT.0) WRITE(6,10000) IFL                           HD2   85
      P = XB(1)                                                              HD2   90
      Q = XB(2)                                                              HD2   95
      RETURN                                                                 HD2  100
10000 FORMAT ('0IFL=',I5)                                                    HD2  105
      END                                                                    HD2  110
```

```
      SUBROUTINE HDGRAD(X,F)                                            HDGR    0
      IMPLICIT REAL*8(A-H,O-Z)                                          HDGR    5
      COMMON VEC(32) ,DMK                                               HDGR   10
      DIMENSION X(1) ,F(1)                                              HDGR   15
      P = X(1)                                                          HDGR   20
      Q = X(2)                                                          HDGR   25
      T1 = 1D0-P-Q                                                      HDGR   30
      T2 = DSQRT(-P*P+2D0*P-2D0*P*Q)                                    HDGR   35
      T3 = DSQRT(-Q*Q+2D0*Q-2D0*P*Q)                                    HDGR   40
      T4 = DSQRT(2D0*Q*P)                                               HDGR   45
      D2 = T2                                                           HDGR   50
      D3 = T3                                                           HDGR   55
      D4 = T4                                                           HDGR   60
      IF (T2.EQ.0D0) D2 = 1D-20                                         HDGR   65
      IF (T3.EQ.0D0) D3 = 1D-20                                         HDGR   70
      IF (T4.EQ.0D0) D4 = 1D-20                                         HDGR   75
      A1 = DSQRT(VEC(1)/DMK)                                            HDGR   80
      A2 = DSQRT(VEC(2)/DMK)                                            HDGR   85
      A3 = DSQRT(VEC(3)/DMK)                                            HDGR   90
      A4 = DSQRT(VEC(4)/DMK)                                            HDGR   95
      UTERM = A1*T1+ A2*T2 + A3*T3 + T4*A4                              HDGR  100
      Y = DSQRT(1D0-UTERM*UTERM)                                        HDGR  105
      IF (Y.EQ.0D0) Y=1D-20                                             HDGR  110
      Z = -1D0/Y                                                        HDGR  115
      F(1) = Z* (-A1 + A2*T1/D2 + A3*(-Q)/D3 + A4*Q/D4)                 HDGR  120
      F(2) = Z*(-A1 + A2*(-P)/D2 + A3*T1/D3 + A4*P/D4)                  HDGR  125
      RETURN                                                            HDGR  130
      END                                                               HDGR  135
```

```
      SUBROUTINE HDVAL(X,F)                                          HDVA   0
      IMPLICIT REAL*8(A-H,O-Z)                                       HDVA   5
      COMMON VEC(32) ,DMK                                            HDVA  10
      DIMENSION X(1)                                                 HDVA  15
      P = X(1)                                                       HDVA  20
      Q = X(2)                                                       HDVA  25
      T1 = 1D0-P-Q                                                   HDVA  30
      T2 = DSQRT(-P*P+2D0*P-2D0*P*Q)                                 HDVA  35
      T3 = DSQRT(-Q*Q+2D0*Q-2D0*P*Q)                                 HDVA  40
      T4 = DSQRT(2D0*Q*P)                                            HDVA  45
      A1 = DSQRT(VEC(1)/DMK)                                         HDVA  50
      A2 = DSQRT(VEC(2)/DMK)                                         HDVA  55
      A3 = DSQRT(VEC(3)/DMK)                                         HDVA  60
      A4 = DSQRT(VEC(4)/DMK)                                         HDVA  65
      UTERM = A1*T1+ A2*T2 + A3*T3 + T4*A4                           HDVA  70
      F = DARCOS(UTERM)                                              HDVA  75
      RETURN                                                         HDVA  80
      END                                                            HDVA  85
```

```
      SUBROUTINE MODCHI(P,Q)                                                MODC   0
      IMPLICIT REAL*8(A-H,O-Z)                                              MODC   5
      COMMON VEC(32),DMK                                                    MODC  10
      COMMON /FAIL/ NFL                                                     MODC  15
      EXTERNAL CONSTR,MODVAL,MODGR                                          MODC  20
      DIMENSION XB(20)                                                      MODC  25
      REAL*8 EPS(20)/2*1D-3/                                                MODC  30
      XB(1) = 1DO-DSQRT((VEC(1)+VEC(3))/DMK)                                MODC  35
      XB(2) = 1DO-DSQRT((VEC(1)+VEC(2))/DMK)                                MODC  40
      IW = 0                                                                MODC  45
      FLR = 0D0                                                             MODC  50
      IF (XB(1)+XB(2).LE.1D0) GO TO 10                                      MODC  55
      XB(1) = .5D0                                                          MODC  60
      XB(2) = .5D0                                                          MODC  65
   10 CALL KEELE(2,6,0,0,5,20,IW,XB,FB,1D-5,1D-7,1D-7,1D-7,1D-7,MODVAL,     MODC  70
     > MODGR ,CONSTR,IFL,FLR,1D-3,EPS)                                      MODC  75
      IF (IW.GT.0) WRITE(6,10000) (VEC(I),I=1,4)                            MODC  80
      IF (IFL.GT.0) NFL = NFL + 1                                           MODC  85
      IF (IFL.GT.0.AND.IW.GT.0) WRITE(6,10100) IFL                          MODC  90
      P = XB(1)                                                             MODC  95
      Q = XB(2)                                                             MODC 100
      RETURN                                                                MODC 105
10000 FORMAT ('0',4F10.5)                                                   MODC 110
10100 FORMAT ('0IFL=',I5)                                                   MODC 115
      END                                                                   MODC 120
```

```
      SUBROUTINE MODVAL(X,F)                                                    MODV    0
      IMPLICIT REAL*8(A-H,O-Z)                                                  MODV    5
      COMMON VEC(32),DMK                                                        MODV   10
      DIMENSION X(1)                                                            MODV   15
      P = X(1)                                                                  MODV   20
      Q = X(2)                                                                  MODV   25
      T1 = 1D0-P-Q                                                              MODV   30
      T2 = -P*P+2D0*P-2D0*P*Q                                                   MODV   35
      T3 = -Q*Q+2D0*Q-2D0*P*Q                                                   MODV   40
      T4 = 2D0*P*Q                                                              MODV   45
      V1 = VEC(1)                                                               MODV   50
      V2 = VEC(2)                                                               MODV   55
      V3 = VEC(3)                                                               MODV   60
      V4 = VEC(4)                                                               MODV   65
      IF (V1.EQ.0D0) V1 = 1D0                                                   MODV   70
      IF (V2.EQ.0D0) V2 = 1D0                                                   MODV   75
      IF (V3.EQ.0D0) V3 = 1D0                                                   MODV   80
      IF (V4.EQ.0D0) V4 = 1D0                                                   MODV   85
      F = (V1 - DMK*T1*T1)**2/V1 + (V2 -DMK*T2)**2/V2 + (V3 -DMK*T3)**2/MODV   90
     > V3 + (V4 -DMK*T4)**2/V4                                                  MODV   95
      RETURN                                                                    MODV  100
      END                                                                       MODV  105
```

```
      SUBROUTINE MODGR(X,F)                                                  MODG   0
      IMPLICIT REAL*8(A-H,O-Z)                                               MODG   5
      DIMENSION X(1),F(1)                                                    MODG  10
      COMMON VEC(32),DMK                                                     MODG  15
      P = X(1)                                                               MODG  20
      Q = X(2)                                                               MODG  25
      T1 = 1D0-P-Q                                                           MODG  30
      T2 = -P*P+2D0*P-2D0*P*Q                                                MODG  35
      T3 = -Q*Q+2D0*Q-2D0*P*Q                                                MODG  40
      T4 = 2D0*P*Q                                                           MODG  45
      V1 = VEC(1)                                                            MODG  50
      V2 = VEC(2)                                                            MODG  55
      V3 = VEC(3)                                                            MODG  60
      V4 = VEC(4)                                                            MODG  65
      IF (V1.EQ.0D0) V1 = 1D0                                                MODG  70
      IF (V2.EQ.0D0) V2 = 1D0                                                MODG  75
      IF (V3.EQ.0D0) V3 = 1D0                                                MODG  80
      IF (V4.EQ.0D0) V4 = 1D0                                                MODG  85
      F(1) = 4D0*(V1 -DMK*T1*T1)*T1*DMK/V1 - 4D0*(V2 -DMK*T2)*T1*DMK/V2      MODG  90
     > + 4D0*(V3 -DMK*T3)*Q*DMK/V3 - 4D0*(V4 -DMK*T4)*Q*DMK/V4               MODG  95
      F(2) = 4D0*(V1 -DMK*T1*T1)*T1*DMK/V1 + 4D0*(V2 -DMK*T2)*P*DMK/V2 -     MODG 100
     >  4D0*(V3 -DMK*T3)*T1*DMK/V3 - 4D0*(V4 -DMK*T4)*P*DMK/V4               MODG 105
      RETURN                                                                 MODG 110
      END                                                                    MODG 115
```

```
      SUBROUTINE CHI2(P,Q)                                                  CHI2   0
      IMPLICIT REAL*8(A-H,O-Z)                                              CHI2   5
      COMMON VEC(32),DMK                                                    CHI2  10
      COMMON /FAIL/ NFL                                                     CHI2  15
      EXTERNAL CONSTR,CH2VAL,CH2GR                                          CHI2  20
      DIMENSION XB(20)                                                      CHI2  25
      REAL*8 EPS(20)/2*1D-3/                                                CHI2  30
      XB(1) = 1DO-DSQRT((VEC(1)+VEC(3))/DMK)                                CHI2  35
      XB(2) = 1DO-DSQRT((VEC(1)+VEC(2))/DMK)                                CHI2  40
      IW = 0                                                                CHI2  45
      FLR = 0DO                                                             CHI2  50
      IF (XB(1)+XB(2).LE.1DO) GO TO 10                                      CHI2  55
      XB(1) = .5DO                                                          CHI2  60
      XB(2) = .5DO                                                          CHI2  65
   10 CALL KEELE(2,6,0,0,5,20,IW,XB,FB,1D-5,1D-7,1D-7,1D-7,1D-7,CH2VAL,     CHI2  70
     > CH2GR ,CONSTR,IFL,FLR,1D-3,EPS)                                      CHI2  75
      IF (IFL.GT.0) NFL = NFL + 1                                           CHI2  80
      IF (IW.GT.0) WRITE(6,10000) (VEC(I),I=1,4)                            CHI2  85
      IF (IFL.GT.0.AND.IW.GT.0) WRITE(6,10100) IFL                          CHI2  90
      P = XB(1)                                                             CHI2  95
      Q = XB(2)                                                             CHI2 100
      RETURN                                                                CHI2 105
10000 FORMAT ('0',4F10.5)                                                   CHI2 110
10100 FORMAT ('0IFL=',I5)                                                   CHI2 115
      END                                                                   CHI2 120
```

```
      SUBROUTINE CH2VAL(X,F)                                            CH2V   0
      IMPLICIT REAL*8(A-H,O-Z)                                          CH2V   5
      COMMON VEC(32),DMK                                                CH2V  10
      DIMENSION X(1)                                                    CH2V  15
      P = X(1)                                                          CH2V  20
      Q = X(2)                                                          CH2V  25
      T1 = 1D0-P-Q                                                      CH2V  30
      T1S = T1*T1                                                       CH2V  35
      T2 = -P*P+2D0*P-2D0*P*Q                                           CH2V  40
      T3 = -Q*Q+2D0*Q-2D0*P*Q                                           CH2V  45
      T4 = 2D0*P*Q                                                      CH2V  50
      F = (VEC(1)-DMK*T1S)**2/(VEC(1) + 2D0) + (VEC(2)-DMK*T2)**2/      CH2V  55
     > (VEC(2) + 2D0) + (VEC(3) - DMK*T3)**2/(VEC(3)+2D0) + (VEC(4)-DMK*CH2V  60
     > T4)**2/(VEC(4)+2D0)                                              CH2V  65
      RETURN                                                            CH2V  70
      END                                                               CH2V  75
```

```
      SUBROUTINE CH2GR (X,F)                                                    CH2G   0
      IMPLICIT REAL*8(A-H,O-Z)                                                  CH2G   5
      DIMENSION X(1),F(1)                                                       CH2G  10
      COMMON VEC(32),DMK                                                        CH2G  15
      P = X(1)                                                                  CH2G  20
      Q = X(2)                                                                  CH2G  25
      T1 = 1D0-P-Q                                                              CH2G  30
      T1S = T1*T1                                                               CH2G  35
      T2 = -P*P+2D0*P-2D0*P*Q                                                   CH2G  40
      T3 = -Q*Q+2D0*Q-2D0*P*Q                                                   CH2G  45
      T4 = 2D0*P*Q                                                              CH2G  50
      V1 = VEC(1) + 2D0                                                         CH2G  55
      V2 = VEC(2) + 2D0                                                         CH2G  60
      V3 = VEC(3) + 2D0                                                         CH2G  65
      V4 = VEC(4) + 2D0                                                         CH2G  70
      U1 = 4D0*(VEC(1)-DMK*T1S)*DMK*T1/V1                                       CH2G  75
      U2 = (VEC(2)-DMK*T2)                                                      CH2G  80
      U3 = (VEC(3)-DMK*T3)                                                      CH2G  85
      U4 = (VEC(4)-DMK*T4)                                                      CH2G  90
      F(1) = U1 - 4D0*U2*DMK*T1/V2 + 4D0*U3*DMK*Q/V3 - 4D0*U4*DMK*Q/V4          CH2G  95
      F(2) = U1 + 4D0*U2*DMK*P/V2 - 4D0*U3*DMK*T1/V3 - 4D0*U4*DMK*P/V4          CH2G 100
      RETURN                                                                    CH2G 105
      END                                                                       CH2G 110
```

```
      SUBROUTINE CHISQ(P,Q)                                              CHIS   0
      IMPLICIT REAL*8(A-H,O-Z)                                           CHIS   5
      COMMON VEC(32) ,DMK                                                CHIS  10
      COMMON /FAIL/ NFL                                                  CHIS  15
      EXTERNAL VALCHI,GRCHI,CONSTR                                       CHIS  20
      DIMENSION XB(20)                                                   CHIS  25
      REAL*8 EPS(20)/2*1D-3/                                             CHIS  30
      XB(1) = 1DO-DSQRT((VEC(1)+VEC(3))/DMK)                             CHIS  35
      XB(2) = 1DO-DSQRT((VEC(1)+VEC(2))/DMK)                             CHIS  40
      FLR = 0DO                                                          CHIS  45
      IW = 0                                                             CHIS  50
      IF (XB(1)+XB(2).LE.1DO) GO TO 10                                   CHIS  55
      XB(1) = .5DO                                                       CHIS  60
      XB(2) = .5DO                                                       CHIS  65
   10 CALL KEELE(2,6,0,0,5,25,IW,XB,FB,1D-5,1D-7,1D-7,1D-7,1D-7,VALCHI,  CHIS  70
     > GRCHI,CONSTR,IFL,FLR,1D-3,EPS)                                    CHIS  75
      IF (IFL.GT.0) NFL = NFL + 1                                        CHIS  80
      IF (IFL.GT.0.AND.IW.GT.0) WRITE(6,10000) IFL                       CHIS  85
      IF (IW.GT.0) WRITE(6,10100) (VEC(I),I=1,4)                         CHIS  90
      P = XB(1)                                                          CHIS  95
      Q = XB(2)                                                          CHIS 100
      RETURN                                                             CHIS 105
10000 FORMAT ('0IFL=',I5)                                                CHIS 110
10100 FORMAT ('0',4F10.5)                                                CHIS 115
      END                                                                CHIS 120
```

```
      SUBROUTINE GRCHI (X,F)                                            GRCH   0
      IMPLICIT REAL*8(A-H,O-Z)                                          GRCH   5
      DIMENSION X(2),F(2)                                               GRCH  10
      COMMON VEC(32),DMK                                                GRCH  15
      P = X(1)                                                          GRCH  20
      Q = X(2)                                                          GRCH  25
      T1 = 1D0-P-Q                                                      GRCH  30
      T2 = -P*P+2D0*P-2D0*P*Q                                           GRCH  35
      T3 = -Q*Q+ 2D0*Q-2D0*Q*P                                          GRCH  40
      T4 = 2D0*P*Q                                                      GRCH  45
      U1 = VEC(1)-DMK*T1*T1                                             GRCH  50
      U2 = VEC(2)-DMK*T2                                                GRCH  55
      U3 = VEC(3)-DMK*T3                                                GRCH  60
      U4 = VEC(4)-DMK*2D0*P*Q                                           GRCH  65
      F1 = 0D0                                                          GRCH  70
      F12 = 0D0                                                         GRCH  75
      F13 = 0D0                                                         GRCH  80
      F14 = 0D0                                                         GRCH  85
      F22 = 0D0                                                         GRCH  90
      F23 = 0D0                                                         GRCH  95
      F24 = 0D0                                                         GRCH 100
      IF (T1.LT.1D-25) GO TO 10                                         GRCH 105
      F1 = 4D0*U1/T1+2D0*U1*U1/(T1**3*DMK)                              GRCH 110
   10 IF (T2.LT.1D-35) GO TO 20                                         GRCH 115
      Z = T2*T2*DMK                                                     GRCH 120
      F12 = -4D0*U2*T1/T2 - 2D0*U2*U2*T1/Z                              GRCH 125
      F22 = 4D0*U2*P/T2 + 2D0*U2*U2*P/Z                                 GRCH 130
   20 IF (T3.LT.1D-35) GO TO 30                                         GRCH 135
      Z = T3*T3*DMK                                                     GRCH 140
      F13 = 4D0*U3*Q/T3 + 2D0*U3*U3*Q/Z                                 GRCH 145
      F23 = -4D0*U3*T1/T3 - 2D0*U3*U3*T1/Z                              GRCH 150
   30 IF (T4.LT.1D-35) GO TO 40                                         GRCH 155
      Z = T4*T4*DMK                                                     GRCH 160
      F14 = -4D0*U4*Q/T4 - 2D0*Q*U4*U4/Z                                GRCH 165
      F24 = -4D0*U4*P/T4 - 2D0*P*U4*U4/Z                                GRCH 170
   40 F(1) = F1 + F12 + F13 + F14                                       GRCH 175
      F(2) = F1 + F22 + F23 + F24                                       GRCH 180
      RETURN                                                            GRCH 185
      END                                                               GRCH 190
```

```
      SUBROUTINE VALCHI (XB,FB)                                          VALC   0
      IMPLICIT REAL*8(A-H,O-Z)                                           VALC   5
      DIMENSION XB(2)                                                    VALC  10
      COMMON VEC(32),DMK                                                 VALC  15
      P = XB(1)                                                          VALC  20
      Q = XB(2)                                                          VALC  25
      A = ((1D0-P-Q)**2)*DMK                                             VALC  30
      D = 2D0*P*Q                                                        VALC  35
      B = (-P*P+2D0*P-D)*DMK                                             VALC  40
      C = (-Q*Q+2D0*Q-D)*DMK                                             VALC  45
      D = D*DMK                                                          VALC  50
      T1 = 0D0                                                           VALC  55
      T2 = 0D0                                                           VALC  60
      T3 = 0D0                                                           VALC  65
      T4 = 0D0                                                           VALC  70
      IF (A.EQ.0D0) GO TO 10                                             VALC  75
      T1 = (VEC(1) - A)**2/A                                             VALC  80
   10 IF (B.EQ.0D0) GO TO 20                                             VALC  85
      T2 = (VEC(2) - B)**2/B                                             VALC  90
   20 IF (C.EQ.0D0) GO TO 30                                             VALC  95
      T3 = (VEC(3) - C)**2/C                                             VALC 100
   30 IF (D.EQ.0D0) GO TO 40                                             VALC 105
      T4 = (VEC(4)-D)**2/D                                               VALC 110
   40 FB = T1 + T2 + T3 + T4                                             VALC 115
      RETURN                                                             VALC 120
      END                                                                VALC 125
```

```
      SUBROUTINE KULL (P,Q)                                                  KULL   0
      IMPLICIT REAL*8(A-H,O-Z)                                               KULL   5
      COMMON VEC(32) ,DMK                                                    KULL  10
      COMMON /FAIL/ NFL                                                      KULL  15
      EXTERNAL GRKULL,VALKUL,CONSTR                                          KULL  20
      DIMENSION XB(20)                                                       KULL  25
      REAL*8 EPS(20)/2*1D-3/                                                 KULL  30
      XB(1) = 1DO-DSQRT((VEC(1)+VEC(3))/DMK)                                 KULL  35
      XB(2) = 1DO-DSQRT((VEC(1)+VEC(2))/DMK)                                 KULL  40
      IW = 0                                                                 KULL  45
      FLR = 0DO                                                              KULL  50
      IF (XB(1)+XB(2).LE.1DO) GO TO 10                                       KULL  55
      XB(1) = .5DO                                                           KULL  60
      XB(2) = .5DO                                                           KULL  65
   10 CALL KEELE(2,6,0,0,5,20,IW,XB,FB,1D-5,1D-7,1D-7,1D-7,1D-7,VALKUL,      KULL  70
     > GRKULL,CONSTR,IFL,FLR,1D-3,EPS)                                       KULL  75
      IF (IFL.GT.0) NFL = NFL + 1                                            KULL  80
      IF (IW.GT.0) WRITE(6,10000) (VEC(I),I=1,4)                             KULL  85
      IF (IFL.GT.0.AND.IW.GT.0) WRITE(6,10100) IFL                           KULL  90
      P = XB(1)                                                              KULL  95
      Q = XB(2)                                                              KULL 100
      RETURN                                                                 KULL 105
10000 FORMAT ('0',4F10.5)                                                    KULL 110
10100 FORMAT ('0IFL=',I5)                                                    KULL 115
      END                                                                    KULL 120
```

```
      SUBROUTINE GRKULL(X,F)                                            GRKU   0
      IMPLICIT REAL*8(A-H,O-Z)                                          GRKU   5
      COMMON VEC(32) ,DMK                                               GRKU  10
      DIMENSION X(1),F(1)                                               GRKU  15
      P = X(1)                                                          GRKU  20
      Q = X(2)                                                          GRKU  25
      T1 = 1D0-P-Q                                                      GRKU  30
      T2 = -P*P+2D0*P-2D0*P*Q                                           GRKU  35
      T3 = -Q*Q +2D0*Q-2D0*P*Q                                          GRKU  40
      T4 = 2D0*P*Q                                                      GRKU  45
      U1 = 0D0                                                          GRKU  50
      U12 = 0D0                                                         GRKU  55
      U22 = 0D0                                                         GRKU  60
      U13 = 0D0                                                         GRKU  65
      U23 = 0D0                                                         GRKU  70
      U14 = 0D0                                                         GRKU  75
      U24 = 0D0                                                         GRKU  80
      IF (VEC(1).EQ.0D0.OR.T1.LE.0D0) GO TO 10                          GRKU  85
      Z = 1D0 + DLOG(DMK/VEC(1)*T1*T1)                                  GRKU  90
      U1 = -2D0*T1*Z                                                    GRKU  95
   10 IF (VEC(2).EQ.0D0.OR.T2.LE.0D0) GO TO 20                          GRKU 100
      Z = 1D0 + DLOG(DMK/VEC(2)*T2)                                     GRKU 105
      U12 = 2D0*T1*Z                                                    GRKU 110
      U22 = -2D0*P*Z                                                    GRKU 115
   20 IF (VEC(3).EQ.0D0.OR.T3.LE.0D0) GO TO 30                          GRKU 120
      Z = 1D0 + DLOG(DMK/VEC(3)*T3)                                     GRKU 125
      U13 = -2D0*Q*Z                                                    GRKU 130
      U23 = 2D0*T1*Z                                                    GRKU 135
   30 IF (VEC(4).EQ.0D0.OR.T4.LE.0D0) GO TO 40                          GRKU 140
      Z = 1D0 + DLOG(DMK/VEC(4)*T4)                                     GRKU 145
      U14 = 2D0*Q*Z                                                     GRKU 150
      U24 = 2D0*P*Z                                                     GRKU 155
   40 F(1) = U1 + U12 + U13 + U14                                       GRKU 160
      F(2) = U1 + U22 + U23 + U24                                       GRKU 165
      RETURN                                                            GRKU 170
      END                                                               GRKU 175
```

```
      SUBROUTINE VALKUL(X,F)                                            VALK   0
      IMPLICIT REAL*8(A-H,O-Z)                                          VALK   5
      COMMON VEC(32) ,DMK                                               VALK  10
      DIMENSION X(1)                                                    VALK  15
      P = X(1)                                                          VALK  20
      Q = X(2)                                                          VALK  25
      T1 = 1D0-P-Q                                                      VALK  30
      T2 = -P*P+2D0*P-2D0*P*Q                                           VALK  35
      T3 = -Q*Q +2D0*Q-2D0*P*Q                                          VALK  40
      T4 = 2D0*P*Q                                                      VALK  45
      U1 = 0D0                                                          VALK  50
      U2 = 0D0                                                          VALK  55
      U3 = 0D0                                                          VALK  60
      U4 = 0D0                                                          VALK  65
      IF (VEC(1).EQ.0D0.OR.T1.LE.0D0) GO TO 10                          VALK  70
      U1 = T1*T1*DLOG(DMK/VEC(1)*T1*T1)                                 VALK  75
   10 IF (VEC(2).EQ.0D0.OR.T2.LE.0D0) GO TO 20                          VALK  80
      U2 = T2*DLOG(DMK/VEC(2)*T2)                                       VALK  85
   20 IF (VEC(3).EQ.0D0.OR.T3.LE.0D0) GO TO 30                          VALK  90
      U3 = T3*DLOG(DMK/VEC(3)*T3)                                       VALK  95
   30 IF (VEC(4).EQ.0D0.OR.T4.LE.0D0) GO TO 40                          VALK 100
      U4 = T4*DLOG(DMK/VEC(4)*T4)                                       VALK 105
   40 F = U1 + U2 + U3 + U4                                             VALK 110
      RETURN                                                            VALK 115
      END                                                               VALK 120
```

```
      SUBROUTINE HAL(P,Q)                                                HAL    0
      IMPLICIT REAL*8(A-H,O-Z)                                           HAL    5
      COMMON VEC(32) ,DMK                                                HAL   10
      COMMON /FAIL/ NFL                                                  HAL   15
      EXTERNAL HALGRA,HALVAL,CONSTR                                      HAL   20
      DIMENSION XB(20)                                                   HAL   25
      REAL*8 EPS(20)/2*1D-3/                                             HAL   30
      XB(1) = 1DO-DSQRT((VEC(1)+VEC(3))/DMK)                             HAL   35
      XB(2) = 1DO-DSQRT((VEC(1)+VEC(2))/DMK)                             HAL   40
      FLR = ODO                                                          HAL   45
      IW = 0                                                             HAL   50
      IF (XB(1)+XB(2).LE.1DO) GO TO 10                                   HAL   55
      XB(1) = .5DO                                                       HAL   60
      XB(2) = .5DO                                                       HAL   65
   10 CALL KEELE(2,6,0,0,5,25,IW,XB,FB,1D-5,1D-7,1D-7,1D-7,1D-7,HALVAL,  HAL   70
     > HALGRA,CONSTR,IFL,FLR,1D-3,EPS)                                   HAL   75
      IF (IFL.GT.0) NFL = NFL + 1                                        HAL   80
      IF (IFL.GT.0.AND.IW.GT.0) WRITE(6,10000) IFL                       HAL   85
      P = XB(1)                                                          HAL   90
      Q = XB(2)                                                          HAL   95
      RETURN                                                             HAL  100
10000 FORMAT ('0IFL=',I5)                                                HAL  105
      END                                                                HAL  110
```

```
      SUBROUTINE HALGRA(X,F)                                                HALG   0
      IMPLICIT REAL*8(A-H,O-Z)                                              HALG   5
      COMMON VEC(32),DMK                                                    HALG  10
      DIMENSION X(1),F(1)                                                   HALG  15
      P = X(1)                                                              HALG  20
      Q = X(2)                                                              HALG  25
      T1 = 1D0-P-Q                                                          HALG  30
      T1CUB = T1**3                                                         HALG  35
      T2 = -P*P + 2D0*P-2D0*P*Q                                             HALG  40
      T3 = -Q*Q + 2D0*Q-2D0*P*Q                                             HALG  45
      V1 = VEC(1) + 1D0                                                     HALG  50
      V2 = VEC(2) + 1D0                                                     HALG  55
      V3 = VEC(3) + 1D0                                                     HALG  60
      V4 = VEC(4) + 1D0                                                     HALG  65
      F(1) =(DMK+1D0)*(-4D0*T1CUB/V1 + 4D0*T2*T1/V2 -4D0*Q*T3/V3 + 8D0*     HALG  70
     > P*Q*Q/V4)                                                            HALG  75
      F(2) = (DMK+1D0)*(-4D0*T1CUB/V1 - 4D0*P*T2/V2 + 4D0*T3*T1/V3 +        HALG  80
     > 8D0*P*P*Q/V4)                                                        HALG  85
      RETURN                                                                HALG  90
      END                                                                   HALG  95
```

```
      SUBROUTINE HALVAL(X,F)                                            HALV   0
      IMPLICIT REAL*8(A-H,O-Z)                                          HALV   5
      COMMON VEC(32),DMK                                                HALV  10
      DIMENSION X(1)                                                    HALV  15
      P = X(1)                                                          HALV  20
      Q = X(2)                                                          HALV  25
      T1 =(1D0-P-Q)**4                                                  HALV  30
      T2 =(-P*P + 2D0*P-2D0*P*Q)**2                                     HALV  35
      T3 =(-Q*Q + 2D0*Q-2D0*P*Q)**2                                     HALV  40
      T4 = (2D0*P*Q)**2                                                 HALV  45
      V1 = VEC(1) + 1D0                                                 HALV  50
      V2 = VEC(2) + 1D0                                                 HALV  55
      V3 = VEC(3) + 1D0                                                 HALV  60
      V4 = VEC(4) + 1D0                                                 HALV  65
      F = (DMK + 1D0)*(T1/V1 + T2/V2 + T3/V3 + T4/V4)                   HALV  70
      RETURN                                                            HALV  75
      END                                                               HALV  80
```

```
      SUBROUTINE MAXLIK(P,Q)                                                MAXL   0
      IMPLICIT REAL*8(A-H,O-Z)                                              MAXL   5
      COMMON VEC(32) ,DMK                                                   MAXL  10
      COMMON /FAIL/ NFL                                                     MAXL  15
      EXTERNAL GRMAXL,VALMAX,CONSTR                                         MAXL  20
      DIMENSION XB(20)                                                      MAXL  25
      REAL*8 EPS(20)/2*1D-3/                                                MAXL  30
      XB(1) = 1DO-DSQRT((VEC(1)+VEC(3))/DMK)                                MAXL  35
      XB(2) = 1DO-DSQRT((VEC(1)+VEC(2))/DMK)                                MAXL  40
      IW =0                                                                 MAXL  45
      FLR = 0DO                                                             MAXL  50
      IF (XB(1)+XB(2).LE.1DO) GO TO 10                                      MAXL  55
      XB(1) = .5DO                                                          MAXL  60
      XB(2) = .5DO                                                          MAXL  65
   10 CALL KEELE(2,6,0,0,5,40,IW,XB,FB,1D-5,1D-7,1D-7,1D-7,1D-7,VALMAX,     MAXL  70
     > GRMAXL,CONSTR,IFL,FLR,1D-3,EPS)                                      MAXL  75
      IF (IFL.GT.0) NFL = NFL + 1                                           MAXL  80
      IF (IFL.GT.0.AND.IW.GT.0) WRITE(6,10000) IFL                          MAXL  85
      IF (IW.GT.0) WRITE(6,10100) (VEC(I),I=1,4)                            MAXL  90
      P = XB(1)                                                             MAXL  95
      Q = XB(2)                                                             MAXL 100
      RETURN                                                                MAXL 105
10000 FORMAT ('0IFL=',I5)                                                   MAXL 110
10100 FORMAT ('0',4F10.5)                                                   MAXL 115
      END                                                                   MAXL 120
```

```
      SUBROUTINE GRMAXL(X,F)                                                GRMA   0
      IMPLICIT REAL*8(A-H,O-Z)                                              GRMA   5
      DIMENSION X(2),F(2)                                                   GRMA  10
      COMMON VEC(32),DMK                                                    GRMA  15
      P = X(1)                                                              GRMA  20
      Q = X(2)                                                              GRMA  25
      T1 = 1D0-P-Q                                                          GRMA  30
      T2 = P                                                                GRMA  35
      T22 = Q                                                               GRMA  40
      T3 = 2D0-P-2D0*Q                                                      GRMA  45
      T4 = 2D0-2D0*P-Q                                                      GRMA  50
      IF (T1.EQ.0D0) T1=1D-20                                               GRMA  55
      IF (T2.EQ.0D0) T2=1D-20                                               GRMA  60
      IF (T3.EQ.0D0) T3=1D-20                                               GRMA  65
      IF (T4.EQ.0D0) T4=1D-20                                               GRMA  70
      IF (T22.EQ.0D0) T22=1D-20                                             GRMA  75
      F(1)= -(-2D0*VEC(1)/T1+(VEC(2)+VEC(4))/T2-VEC(2)/T3-2D0*VEC(3)/T4)GRMA  80
      F(2)=-(-2D0*VEC(1)/T1+(VEC(3)+VEC(4))/T22-2D0*VEC(2)/T3-VEC(3)/T4)GRMA  85
      RETURN                                                                GRMA  90
      END                                                                   GRMA  95
```

```
      SUBROUTINE VALMAX(XB,FB)                                          VALM   0
      IMPLICIT REAL*8(A-H,O-Z)                                          VALM   5
      COMMON VEC(32),DMK                                                VALM  10
      DIMENSION XB(2)                                                   VALM  15
      COMMON/LOGFAC/FL(201)                                             VALM  20
      P = XB(1)                                                         VALM  25
      Q = XB(2)                                                         VALM  30
      CON=FL(DMK+1)-FL(VEC(1)+1)-FL(VEC(2)+1)-FL(VEC(3)+1)-FL(VEC(4)+1) VALM  35
      T1 = 1D0-P-Q                                                      VALM  40
      T2 = P                                                            VALM  45
      T3 = Q                                                            VALM  50
      T4 = 2D0-P-Q*2D0                                                  VALM  55
      T5 = 2D0-2D0*P-Q                                                  VALM  60
      IF (T1.LE.0D0) T1 = 1D-20                                         VALM  65
      IF (T2.LE.0D0) T2 = 1D-20                                         VALM  70
      IF (T3.LE.0D0) T3 = 1D-20                                         VALM  75
      IF (T4.LE.0D0) T4 = 1D-20                                         VALM  80
      IF (T5.LE.0D0) T5 = 1D-20                                         VALM  85
      FB=-(CON + 2D0*VEC(1)*DLOG(T1) + (VEC(2) + VEC(4))*DLOG(T2) +     VALM  90
     > (VEC(3) + VEC(4))*DLOG(T3) + VEC(2)*DLOG(T4) + VEC(3)*DLOG(T5))  VALM  95
      RETURN                                                            VALM 100
      END                                                               VALM 105
```

```
      SUBROUTINE BERN(P,Q)                                                  BERN    0
      IMPLICIT REAL*8(A-H,O-Z)                                              BERN    5
      COMMON VEC(32),DMK                                                    BERN   10
      COMMON/RHAT/R                                                         BERN   15
      P = 1DO-DSQRT((VEC(1) + VEC(3))/DMK)                                  BERN   20
      Q = 1DO-DSQRT((VEC(1) + VEC(2))/DMK)                                  BERN   25
      R = DSQRT(VEC(1)/DMK)                                                 BERN   30
      IW = 1                                                                BERN   35
      IF (IW.GT.0) WRITE(6,10000) (VEC(I),I=1,4),P,Q,R                      BERN   40
      RETURN                                                                BERN   45
10000 FORMAT ('0',4F10.0/' ',3D16.8)                                        BERN   50
      END                                                                   BERN   55
```

```
      SUBROUTINE WEINER(P,Q)                                            WEIN   0
      IMPLICIT REAL*8(A-H,O-Z)                                          WEIN   5
      COMMON VEC(32),DMK                                                WEIN  10
      COMMON/RHAT/R                                                     WEIN  15
      R = DSQRT(VEC(1)/DMK)                                             WEIN  20
      P = DSQRT((VEC(1) + VEC(2))/DMK) - R                              WEIN  25
      Q = DSQRT((VEC(1) + VEC(3))/DMK) - R                              WEIN  30
      IW = 0                                                            WEIN  35
      IF (IW.GT.0) WRITE(6,10000) (VEC(I),I=1,4),P,Q,R                  WEIN  40
      RETURN                                                            WEIN  45
10000 FORMAT ('0',4F10.0/' ',3D16.8)                                    WEIN  50
      END                                                               WEIN  55
```

```
      SUBROUTINE IMPROV(P,Q)                                            IMPR    0
      IMPLICIT REAL*8(A-H,O-Z)                                          IMPR    5
      COMMON VEC(32),DMK                                                IMPR   10
      COMMON/RHAT/R                                                     IMPR   15
      PP= 1DO-DSQRT((VEC(1) + VEC(3))/DMK)                              IMPR   20
      QP= 1DO-DSQRT((VEC(1) + VEC(2))/DMK)                              IMPR   25
      RP = DSQRT(VEC(1)/DMK)                                            IMPR   30
      D = 1DO-PP-QP-RP                                                  IMPR   35
      D2 = .5DO*D                                                       IMPR   40
      DP1 = 1DO+ D2                                                     IMPR   45
      P = DP1*PP                                                        IMPR   50
      Q = DP1*QP                                                        IMPR   55
      R = DP1*(RP+D2)                                                   IMPR   60
      IW = 0                                                            IMPR   65
      IF (IW.GT.0) WRITE(6,10000) (VEC(I),I=1,4),P,Q,R                  IMPR   70
      RETURN                                                            IMPR   75
10000 FORMAT ('0',4F10.0/' ',3D16.8)                                    IMPR   80
      END                                                               IMPR   85
```

```
      IMPLICIT REAL*8(A-H,O-Z)                                          MAIN    0
      REAL*8 DMK(6)/30D0,35D0,40D0,45D0,50D0,75D0/                      MAIN    5
      REAL*8 PP(10)/.05D0,.15D0,.25D0,.33D0,.4D0,.45D0,.55D0,.65D0,     MAIN   10
     > .75D0,.85D0/                                                     MAIN   15
      REAL*8 QQ(10)/.05D0,.15D0,.25D0,.33D0,.40D0,.45D0,.25D0,.15D0,    MAIN   20
     > .10D0,.05D0/                                                     MAIN   25
      K = 0                                                             MAIN   30
   10 K = K + 1                                                         MAIN   35
      IF (K.GT.3) GO TO 140                                             MAIN   40
      P = 1D0/3D0                                                       MAIN   45
      Q = 1D0/3D0                                                       MAIN   50
      R = 1D0-P-Q                                                       MAIN   55
      GO TO (20,30,40),K                                                MAIN   60
   20 WRITE(6,10000)                                                    MAIN   65
      GO TO 50                                                          MAIN   70
   30 WRITE(6,10100)                                                    MAIN   75
      GO TO 50                                                          MAIN   80
   40 WRITE(6,10200)                                                    MAIN   85
   50 CONTINUE                                                          MAIN   90
      DO 90 I=1,6                                                       MAIN   95
         GO TO (60,70,80),K                                             MAIN  100
   60    CALL MAX(P,Q,R,DMK(I))                                         MAIN  105
         GO TO 90                                                       MAIN  110
   70    CALL BERN(P,Q,R,DMK(I))                                        MAIN  115
         GO TO 90                                                       MAIN  120
   80    CALL WIENER(P,Q,R,DMK(I))                                      MAIN  125
   90    CONTINUE                                                       MAIN  130
      DO 130 I=1,10                                                     MAIN  135
         P = PP(I)                                                      MAIN  140
         IF (P.EQ..33D0) P = 1D0/3D0                                    MAIN  145
         Q = QQ(I)                                                      MAIN  150
         IF (Q.EQ..33D0) Q = 1D0/3D0                                    MAIN  155
         R = 1D0-P-Q                                                    MAIN  160
         GO TO (100,110,120),K                                          MAIN  165
  100    CALL MAX(P,Q,R,50D0)                                           MAIN  170
         GO TO 130                                                      MAIN  175
  110    CALL BERN(P,Q,R,50D0)                                          MAIN  180
         GO TO 130                                                      MAIN  185
  120    CALL WIENER(P,Q,R,50D0)                                        MAIN  190
  130    CONTINUE                                                       MAIN  195
      GO TO 10                                                          MAIN  200
  140 RETURN                                                            MAIN  205
10000 FORMAT ('1MAXIMUM LIKELIHOOD MATRIX OF COVARIANCES')              MAIN  210
10100 FORMAT ('1  BERNSTEIN''S METHOD MATRIX OF COVARIANCES')           MAIN  215
10200 FORMAT ('1WIENER''S METHOD   MATRIX OF COVARIANCES')              MAIN  220
      END                                                               MAIN  225
```

```
      SUBROUTINE WIENER(P,Q,R,DMK)                                          WIEN   0
      IMPLICIT REAL*8(A-H,O-Z)                                              WIEN   5
      DIMENSION A(2,2)                                                      WIEN  10
      C = .5D0/DMK                                                          WIEN  15
      D = .25D0/DMK                                                         WIEN  20
      A(1,1) = C*P*(1D0-P) + D*P*P*(1D0+2D0*Q/(P*P+P*R))                    WIEN  25
      A(2,2) = C*Q*(1D0-Q) + D*Q*Q*(1D0 + 2D0*P/(Q*Q+Q*R))                  WIEN  30
      A(1,2) = -C*P*Q + D*P*Q*(1D0 + 1D0/(P*Q+R))                           WIEN  35
      A(2,1) = A(1,2)                                                       WIEN  40
      WRITE(6,10000) P,Q,R,DMK                                              WIEN  45
      WRITE(6,10100) A                                                      WIEN  50
      RETURN                                                                WIEN  55
10000 FORMAT (' '/' P=',F6.3,2X,'Q=',F6.3,2X,'R=',F6.3,2X,'N=',F4.0)        WIEN  60
10100 FORMAT ('0',2D16.8/' ',2D16.8)                                        WIEN  65
      END                                                                   WIEN  70
```

```
      SUBROUTINE BERN(P,Q,R,DMK)                                        BERN   0
      IMPLICIT REAL*8(A-H,O-Z)                                          BERN   5
      DIMENSION A(2,2)                                                  BERN  10
      T = 1D0-R/(P*Q+R)                                                 BERN  15
      C = .5D0/DMK                                                      BERN  20
      D = .25D0/DMK                                                     BERN  25
      A(1,1) = C*P*(1D0-P) + D*P*P                                      BERN  30
      A(2,2) = C*Q*(1D0-Q) + D*Q*Q                                      BERN  35
      A(1,2) = -C*P*Q + D*P*Q*T                                         BERN  40
      A(2,1) = A(1,2)                                                   BERN  45
      WRITE(6,10000) P,Q,R,DMK                                          BERN  50
      WRITE(6,10100) A                                                  BERN  55
      RETURN                                                            BERN  60
10000 FORMAT (' '/' P=',F6.3,2X,'Q=',F6.3,2X,'R=',F6.3,2X,'N=',F4.0)    BERN  65
10100 FORMAT ('0',2D16.8/' ',2D16.8)                                    BERN  70
      END                                                               BERN  75
```

```
      SUBROUTINE MAX(P,Q,R,DMK)                                          MAX     0
      IMPLICIT REAL*8(A-H,O-Z)                                           MAX     5
      DIMENSION A(2,2)                                                   MAX    10
      C = .5D0/DMK                                                       MAX    15
      D = .125D0/DMK                                                     MAX    20
      T = R/(P*Q+R)                                                      MAX    25
      A(1,1) = C*P*(1D0-P) + D*P*P*(1D0+T)                               MAX    30
      A(2,2) = C*Q*(1D0-Q) + D*Q*Q*(1D0+T)                               MAX    35
      A(1,2) = -C*P*Q + D*P*Q*(1D0-T)                                    MAX    40
      A(2,1) = A(1,2)                                                    MAX    45
      WRITE(6,10000) P,Q,R,DMK                                           MAX    50
      WRITE(6,10100) A                                                   MAX    55
      RETURN                                                             MAX    60
10000 FORMAT (' '/' P=',F6.3,2X,'Q=',F6.3,2X,'R=',F6.3,2X,'N=',F4.0)     MAX    65
10100 FORMAT ('0',2D16.8/' ',2D16.8)                                     MAX    70
      END                                                                MAX    75
```

APPENDIX B

COMPUTER PROGRAM FOR PROBABILITY SPACE FOR MULTINOMIAL DISTRIBUTION OF k CATEGORIES AND n TRIALS

Purpose

The algorithm generates an orderly listing of all distinct partitions and permutations of n into k categories.

The algorithm is useful in problems of estimation of the multinomial distribution (see for example Bowman and Shenton [1971] and Bowman, Hutcheson, Odum, and Shenton [1970, pp. 315-366]. A multinomial distribution is specified by a vector $\underline{p}:\{p_1,\ldots,p_k\}$ of probabilities of k mutually exclusive events. If n independent trials yield n_1 events of the first kind, n_2 events of the second kind, and so on, the probability of $n_1,\ldots,n_k$ given $p_1,\ldots,p_k$ is given in equation (B.1).

$$\text{(B.1)} \qquad \frac{n!}{n_1!\ldots n_k!}\, p_1^{n_1}\ldots p_k^{n_k}$$

The algorithm generates all possible sample configurations $(n_1.,\ldots,n_k)$, with n_i's being nonnegative integers summing to n. The total number of distinct partitions and permutations is $\left[\begin{smallmatrix} n+k-1 \\ k-1 \end{smallmatrix}\right]$. If the k elements of the vector $(n_1,\ldots,n_k)$ are taken to be a number, the algorithm generates the numbers in ascending order.

Method

We will examine the case of n equal to 3 and k equal to 4. There are $\left[\begin{smallmatrix} n+k-1 \\ k-1 \end{smallmatrix}\right] = 20$ distinct partitions and permutations of 3

into 4 categories. The algorithm will generate the vectors in ascending order as shown in Table 22.

The first vector has value of 0 in elements 1 through 3 and value of n, or 3, in the k^{th}, or fourth, element. It can be seen in Table 22 that vectors 1 through 4 contain distinct partitions and permutations of 3 into 2 categories in elements 3 and 4 of the vectors. For the fifth vector, element 2 is incremented by 1. For vectors 5 through 7, all distinct partitions and permutations of 2 into 2 categories are in elements 3 and 4 of these vectors. Element 2 is again incremented by 1 and the process continued until we see in vector 10 that element 2 is equal to n, or 3. Now by Table 22 it is seen that elements 2, 3, and 4 of vectors 1 through 10 contain all distinct partitions and permutations of 3 into 3 categories. For vector 11, element 1 is incremented by 1. The process is continued until element 1 is equal to n, or 3.

How to use

The driver program determines the value of n and k. For the first call to subroutine Space, the argument IENT has value 0. The array A will contain the first vector of k elements upon return to the driver program. For all following calls to subroutine Space the argument IENT should have the value of 1. The array A will contain the next vector upon return from each call. The argument IRET will have the value of 0 upon return from subroutine Space if A contains another vector; IRET will have the value of 1 if there are no more vectors. The driver program should not change the value of A between calls.

TABLE 22. All distinct partitions and permutations of 3 into 4 categories.

I	V(I)	Elements 3 and 4 contain partitions and permutations of	Elements 2, 3, and 4 contain partitions and permutations of
1	0003	3 into 2 categories	3 into 3 categories
2	0012		
3	0021		
4	0030		
5	0102	2 into 2 categories	
6	0111		
7	0120		
8	0201	1 into 2 categories	
9	0210		
10	0300	0 into 2 categories	
11	1002	2 into 2 categories	2 into 3 categories
12	1011		
13	1020		
14	1101	1 into 2 categories	
15	1110		
16	1200	0 into 2 categories	
17	2001	1 into 2 categories	1 into 3 categories
18	2010		
19	2100	0 into 2 categories	
20	3000	0 into 2 categories	0 into 3 categories

The current capacity of the program is n greater than or equal to 1 and k between 2 and 200. If k greater than 200 is desired, change the dimension of IS in subroutine Space to the value of k-1.

A listing of the program is in Figure 12.

Time

The author ran the case of n equal to 100 and k equal to 4 on an IBM 360/65 computer. The 176,851 solution vectors were found in a central processing time of 22.15 seconds. This time included execution of subroutine Space and the driver program and a minimum of output.

Structure

Table 23 describes the structure of the subroutine Space, which has arguments A, N, K, IENT, and IRET.

```
      SUBROUTINE SPACE(A,N,K,IENT,IRET)                                  SPAC   0
      INTEGER A(K),IS(199)                                               SPAC   5
      IF (IENT.EQ.1) GO TO 20                                            SPAC  10
C *** ENTER HERE FOR FIRST CALL                                          SPAC  15
      IRET = 0                                                           SPAC  20
      KM1 = K-1                                                          SPAC  25
      DO 10 I=1,KM1                                                      SPAC  30
         A(I) = 0                                                        SPAC  35
   10    IS(I) = N                                                       SPAC  40
      A(K) = N                                                           SPAC  45
      RETURN                                                             SPAC  50
C *** ENTER HERE FOR ALL CALLS AFTER FIRST                               SPAC  55
   20 CONTINUE                                                           SPAC  60
      IF (A(KM1).EQ.IS(1)) GO TO 30                                      SPAC  65
      A(KM1) = A(KM1) + 1                                                SPAC  70
      A(K) = A(K) - 1                                                    SPAC  75
      RETURN                                                             SPAC  80
   30 CONTINUE                                                           SPAC  85
      IF (KM1.EQ.1) GO TO 50                                             SPAC  90
      DO 40 I=2,KM1                                                      SPAC  95
         IF (A(K-I).EQ.IS(I)) GO TO 40                                   SPAC 100
         J = I                                                           SPAC 105
         GO TO 60                                                        SPAC 110
   40    CONTINUE                                                        SPAC 115
   50 IRET = 1                                                           SPAC 120
C *** THERE ARE NO MORE SOLUTIONS                                        SPAC 125
      RETURN                                                             SPAC 130
   60 CONTINUE                                                           SPAC 135
      M = K-J                                                            SPAC 140
      A(M) = A(M) + 1                                                    SPAC 145
      L = K-J+1                                                          SPAC 150
      DO 70 I=L,KM1                                                      SPAC 155
   70    A(I) = 0                                                        SPAC 160
      A(K) = N                                                           SPAC 165
      DO 80 I=1,M                                                        SPAC 170
   80    A(K) = A(K) - A(I)                                              SPAC 175
      JM1 = J-1                                                          SPAC 180
      DO 90 I=1,JM1                                                      SPAC 185
   90    IS(I) = A(K)                                                    SPAC 190
      RETURN                                                             SPAC 195
      END                                                                SPAC 200
```

FIGURE 12. Listing of subroutine Space.

TABLE 23. Structure of subroutine Space.

Parameters	Type		Description
A(k)	integer array	output:	contains each vector
N	integer	input:	sample size
K	integer	input:	number of categories
IENT	integer	input:	has value of 0 for first call to Space; has value of 1 for all succeeding calls to Space
IRET	integer	output:	has value of 1 if this is final exit; has value of 0 otherwise
IS(199)	integer	workspace:	contains counters

VITA

Barbara Lynne Jackson Handley was born in Knoxville, Tennessee, on August 19, 1947. She attended elementary schools in that city and was graduated from South High School in 1965. In June, 1965, she entered The University of Tennessee; and in June, 1968, she received a Bachelor of Science degree in Statistics. Since June, 1968, she has been employed by the Computing Technology Center, Union Carbide Corporation, Oak Ridge, Tennessee.

In September, 1968, she entered the Graduate School at The University of Tennessee and received the Master of Science degree in Mathematics in March, 1973. She is a member of the American Statistical Association and Phi Kappa Phi.

She is married to Gary Richard Handley, formerly of Huntington, West Virginia.