ESTIMATING COMPONENT FAILURE RATES AND YIELDS

ESTIMATION METHODS FOR DETERMINING FAILURE RATES OF COMPONENT FAMILIES FROM OBSERVED FAILURE RATES OF UNITS

AND

YIELDS OF COMPONENT FAMILIES FROM OBSERVED YIELDS OF UNITS

By

LAWRENCE WESLEY STITT, B.Sc.

A Project

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Science

McMaster University

December 1976

MASTER OF SCIENCE (1976) McMaster University
(Statistics) Hamilton, Ontario

TITLE: Estimation Methods for Determining Failure Rates of Component Families from Observed Failure Rates of Units and Yields of Component Families from Observed Yields of Units

AUTHOR: Lawrence Wesley Stitt, B.Sc. (McMaster University)

SUPERVISORS: Professor P.D.M. Macdonald (McMaster University)
Mr. Antonin Wild (Bell Northern Research)

NUMBER OF PAGES: viii, 100

ABSTRACT

In the field of communications electronic plug-in units operating together form a system. In the event of failure a plug-in unit can easily be replaced by another. Each unit consists of electronic components soldered onto a board in a particular pattern. A component may be either a single electronic part such as a transistor or a combination of single parts such as an integrated circuit. Electronic components with similar properties have been grouped into families. This reduces the number of parameters to be estimated from the observations available.

The method of maximum likelihood is used to estimate the failure rates of component families. The number of unit failures and the number of units in use observed during measured periods of time and the component family makeup of the observed units are used to make the estimates.

The probability of an electronic component from a given component from a given family being acceptable after the production process will be referred to as the component yield for that family. Similarly, the probability of a given type of unit being acceptable after the production process will be referred to as the yield for that type of unit. By taking the logarithms of the yields, the estimation problem can be reduced to the linear problem of estimating logarithms of component family yields. Using unit yields, the total

number of each type of unit produced, and the component family makeup of those units produced, component family yields are estimated. The method of maximum likelihood is applied directly to the data and the method of weighted least squares is applied to the linearized problem.

ACKNOWLEDGEMENTS

I wish to thank both Dr. P.D.M. Macdonald of McMaster University and Mr. Antonin Wild of Bell Northern Research for their assistance in this work.

A special thanks is due to Mr. Diju Raha of Bell Northern Research for providing the problems and the resources of Bell Northern Research with which to work towards solutions. Thanks are also due to other employees of Bell Northern Research including Mr. John Rakos, Mr. Attilio DeBernardo and Miss Pat Paley, for their assistance in times of need.

Finally, I wish to thank Susan Baker for typing the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

One of the tasks of a reliability engineer in the communications industry is prediction. If the yield of electronic units from the production line can be predicted, producers can better predict production costs and hence determine a selling price for their products. If the reliability of a newly designed electronic unit can be predicted, purchasers of the units can be given estimates of their maintenance costs. Knowledge of the component family yields and component family reliabilities would make these predictions possible.

Components within an electronic unit may either be in series or in parallel. In the case of components in series, a component failure will cause a break in the circuit thereby resulting in a unit failure. When a component which is parallel with another fails the circuit is not broken and hence the unit continues to function. Parallel components are sometimes installed in order to lengthen the lifetime of an electronic circuit. Designers of electronic plug-in units tend to avoid parallel components because a plug-in unit can be easily replaced, parallel components add to the cost of production, and the lifetime of a unit is not greatly prolonged by the addition of extra components. It is therefore reasonable to assume that components within units are in series.

CHAPTER 2

FAILURE RATES OF COMPONENT FAMILIES FROM UNIT FAILURE RATES

2.1 Introduction

The reliability of an electronic unit can be estimated if the unit has been in service for a length of time, but no adequate method has been developed for predicting the reliability of a newly designed unit. If estimates of component family reliabilities were available this would be possible. When units from the field are serviced, a list of those components replaced can be obtained. Unfortunately no differentiation is made between those components which have been removed because of failure and those removed as trouble shooting aids. As a result, no direct method of estimating component reliabilities is available. Testing of the reliability of electronic components under actual operating conditions is costly in terms of both time and money. Accelerated testing, where components are tested at their highest ratings of temperature and humidity have been done. However, in the tests, chemical reactions which occur within components at different temperatures and humidities make the results difficult to apply. Tests of the reliability of electronic components have been done in the aircraft industry but cannot be applied because of the effect of vibration as well as different temperature and humidity conditions. For purposes of prediction, a method of estimating component family

reliabilities is needed by reliability engineers.

Units within a system can be assumed to operate independently. The failure of one unit does not damage another unit or cause another unit to fail. The same assumption can be made for individual components since those components which have failed or have been damaged as a result of the original component failure are replaced along with the original component which failed. Unfortunately damage to components cannot always be detected. Component failures are assumed to follow a Poisson process with a constant failure rate. With a constant failure rate, components can be replaced without affecting the model.

2.2 Method of Maximum Likelihood

Consider N distinct types of electronic units produced using components from M distinct component families. Since components are Poisson:

$$P[\,z \text{ components of type } j \text{ failing in time } t] = \frac{e^{-\alpha_j t}(\alpha_j t)^z}{z!} \quad (j=1,\ldots,M),$$

where α_j is the constant failure rate of a component of type j. Since component failures are independent and components are in series:

$$\lambda_i = \sum_{j=1}^{M} \alpha_j \, q_{ij} \qquad (i=1,\ldots, N),$$

where λ_i is the constant failure rate of a unit of type i and q_{ij} is the number of components of type j in a unit of type i.

The total number of units of a particular type in actual use and the total number in the field may vary from one time period to

another. As a result, the expected number of failures may differ from one period to another. Let Y_{ik} be the random variable denoting the number of failures of a unit of type i during the k^{th} time period. Then Y_{ik} is Poisson with parameter ϕ_{ik} where:

$$\phi_{ik} = \gamma_{ik}\, n_{ik}\, t_k\, \lambda_i \qquad (i=1,\ldots,N;\ k=1,\ldots,K);$$

and γ_{ik} is the proportion of the units of type i in the field actually used during the k^{th} time period, n_{ik} is the total number of units of type in the field during the k^{th} time period, and t_k is thelength of the k^{th} time period.

The likelihood function L can now be set up:

$$L(\alpha_1,\ldots,\alpha_M|y_{11},\ldots,y_{NK}) = \prod_{i=1}^{N}\prod_{k=1}^{K} \frac{e^{-\phi_{ik}}\,\phi_{ik}^{\,y_{ik}}}{y_{ik}!}.$$

Maximizing the likelihood is equivalent to maximizing the logarithm of the likelihood:

$$\log L(\alpha_1,\ldots,\alpha_M|y_{11},\ldots,y_{NK}) = \sum_{i=1}^{N}\sum_{k=1}^{K}\{y_{ik}\log\phi_{ik} - \phi_{ik} - \log(y_{ik}!)\}.$$

To maximize logL, its derivatives are taken with respect to the parameters $\alpha_s (s=1,\ldots,M)$ and set equal to zero. Since:

$$\phi_{ik} = \gamma_{ik}\, n_{ik}\, t_k \sum_{j=1}^{M} \alpha_j\, q_{ij} \qquad (i=1,\ldots,N;\ k=1,.,K),$$

it follows that:

$$\frac{\partial \phi_{ik}}{\partial \alpha_s} = \gamma_{ik}\, n_{ik}\, t_k\, q_{is} \qquad (i=1,\ldots,N;\ k=1,\ldots,K;\ s=1,\ldots,M).$$

Hence:

$$\frac{\partial \log L}{\partial \alpha_s} = \sum_{i=1}^{N} \sum_{k=1}^{K} \left\{ \frac{y_{ik}}{\phi_{ik}} \frac{\partial \phi_{ik}}{\partial \alpha_s} - \frac{\partial \phi_{ik}}{\partial \alpha_s} \right\}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} \left\{ \gamma_{ik}\, n_{ik}\, t_k\, q_{is} \left\{ \frac{y_{ik}}{\phi_{ik}} - 1 \right\} \qquad (s=1,\ldots,M).$$

Set derivatives equal to zero and solve for $\hat{\alpha}_s$ $(s=1,\ldots,M)$ where $\hat{\alpha}_s$ is the maximum likelihood estimate of the failure rate of a component of the s^{th} component family:

$$\sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik}\, n_{ik}\, t_k\, q_{is} \left\{ \frac{y_{ik}}{\gamma_{ik}\, n_{ik}\, t_k \sum_{j=1}^{M} \hat{\alpha}_j\, q_{ij}} - 1 \right\} = 0 \qquad (s=1,\ldots,M).$$

The method of scoring (Appendix 1) is used to solve this set of non-linear equations. The information matrix is required for scoring:

$$\frac{\partial^2 \log L}{\partial \alpha_t\, \partial \alpha_s} = \sum_{i=1}^{N} \sum_{k=1}^{K} q_{is}\, y_{ik} \frac{\partial}{\partial \alpha_t} \left(\frac{1}{\sum_{j=1}^{M} \alpha_j\, q_{ij}} \right)$$

$$= - \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{y_{ik}\, q_{is}\, q_{it}}{\left(\sum_{j=1}^{M} \alpha_j\, q_{ij} \right)^2} \qquad (s=1,\ldots,M;\ t=1,\ldots,M),$$

$$E\left[\frac{-\partial^2 \log L}{\partial \alpha_t\, \partial \alpha_s} \right] = \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{\gamma_{ik}\, n_{ik}\, t_k\, q_{is}\, q_{it}}{\sum_{j=1}^{M} \alpha_j\, q_{ij}} = \vartheta(s,t)$$

$$(s=1,\ldots,M;\ t=1,\ldots,M),$$

$$\hat{\vartheta}(s,t) = \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{\gamma_{ik}\, n_{ik}\, t_k\, q_{is}\, q_{it}}{\sum_{j=1}^{M} \hat{\alpha}_j\, q_{ij}} \qquad (s=1,\ldots,M;\ t=1,\ldots,M).$$

Using the inverse of the estimated information matrix an approximate $100(1-\varepsilon)\%$ confidence interval for the s^{th} component family failure rate is

$$\hat{\alpha}_s \pm z_{\varepsilon/2} \sqrt{\hat{\vartheta}^{-1}(s,s)} \qquad (s=1,\ldots,M),$$

where $z_{\varepsilon/2}$ is the $100\ \varepsilon/2\%$ critical value for the normal distribution and $\hat{\vartheta}^{-1}(s,s)$ is the (s,s) entry of the inverse of the estimated information matrix.

Unit failure rates can now be predicted using our estimated component family failure rates:

$$\hat{\lambda}_0 = \sum_{j=1}^{M} \hat{\alpha}_j\, q_{0j} \ ,$$

where $\hat{\lambda}_0$ is the predicted failure rate of any unit made of components from the M component families for which we estimated failure rates and q_{0j} is the number of components of type j in that unit.

$$\begin{aligned} \mathrm{Var}(\hat{\lambda}_0) &= \mathrm{Var}\Big(\sum_{j=1}^{M} \hat{\alpha}_j\, q_{0j}\Big) \\ &= \sum_{k=1}^{M} \sum_{j=1}^{M} q_{0j}\, q_{0k}\, \mathrm{Cov}(\hat{\alpha}_j, \hat{\alpha}_k) \\ &\simeq \sum_{k=1}^{M} \sum_{j=1}^{M} q_{0j}\, q_{0k}\, \vartheta^{-1}(j,k) \\ &\simeq \sum_{k=1}^{M} \sum_{j=1}^{M} q_{0j}\, q_{0k}\, \hat{\vartheta}^{-1}(j,k) \end{aligned}$$

Then approximate 100(1-ε)% confidence intervals for the unit failure rates based on the predicted failure rates are:

$$\hat{\lambda}_0 \pm z_{\varepsilon/2} \sqrt{\sum_{k=1}^{M} \sum_{j=1}^{M} q_{0j}\, q_{0k}\, \hat{\mathcal{J}}^{-1}(j,k)}$$

Using the normal approximation to the Poisson distribution, approximate 100(1-ε)% confidence intervals for the unit failure rates based on the observed failure rates are:

$$\sum_{k=1}^{K} \frac{y_{ik}}{\gamma_{ik}\, n_{ik}\, t_k} \pm z_{\varepsilon/2} \sqrt{\sum_{k=1}^{K} \frac{y_{ik}}{\gamma_{ik}\, n_{ik}\, t_k}} \qquad (i=1,\ldots,N).$$

For those units used to estimate the component family failure rates, the approximate confidence intervals for unit failure rates based on predicted failure rates can be compared with those based on observed failure rates.

An approximate test of fit can be done using the normal approximation to the Poisson distribution. The statistic

$\sum_{i=1}^{N} \sum_{k=1}^{K} \frac{(y_{ik} - \hat{\phi}_{ik})^2}{\hat{\phi}_{ik}}$ is asymptotically Chi squared on N·K-M degrees of freedom.

2.3 Discussion and Recommendations

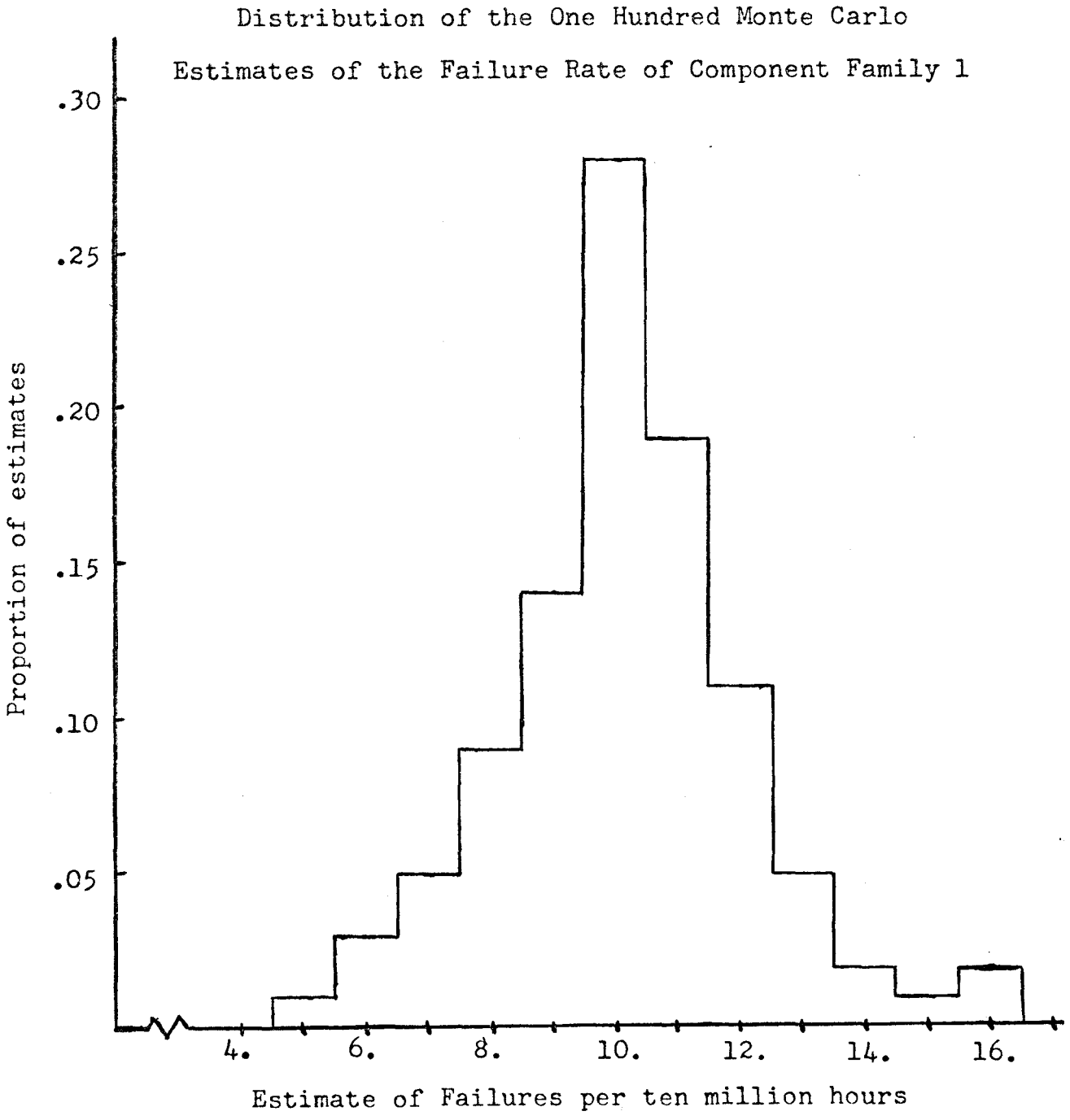
2.3.1 Monte Carlo Study:

Ten distinct artificial units, each made of components from three distinct families with known failure rates, were simulated. Setting the ratio of units in use to the number of units in the field to one, the number of each type of unit in use for a period of five

thousand hours was chosen to represent possible actual conditions. One hundred samples of the number of failures of each of the ten units were randomly generated using the normal approximation to the Poisson distribution. If Y_i is Poisson with mean ϕ_i and variance ϕ_i, then Y_i is approximately normal with mean ϕ_i and variance ϕ_i. Estimates of the component family failure rates and predicted unit failure rates were then made for the one hundred samples using the method of maximum likelihood.

Approximate ninety-five percent confidence intervals for each of the component family failure rate estimates for the one hundred samples were calculated under the assumption that the component family failure rates are approximately normal. The estimates made from the generated unit failure rate data appear to be approximately normal (Figures 1, 2 and 3). Under actual conditions, it can be expected that the same will occur. A likelihood ratio test was used to test equality of the estimated variance-covariance matrix of the one hundred sets of component family failure rates and the inverse of the true information matrix (T.W. Anderson, 1958, Section 10.8). The calculated statistic was 5.16. The five percent Chi-squared critical value on six degrees of freedom is 12.6, meaning that the hypothesis of equality of the two matrices is accepted at the five percent level. Then the use of the inverse of the estimated information matrix in calculating confidence intervals should not affect the level of confidence.

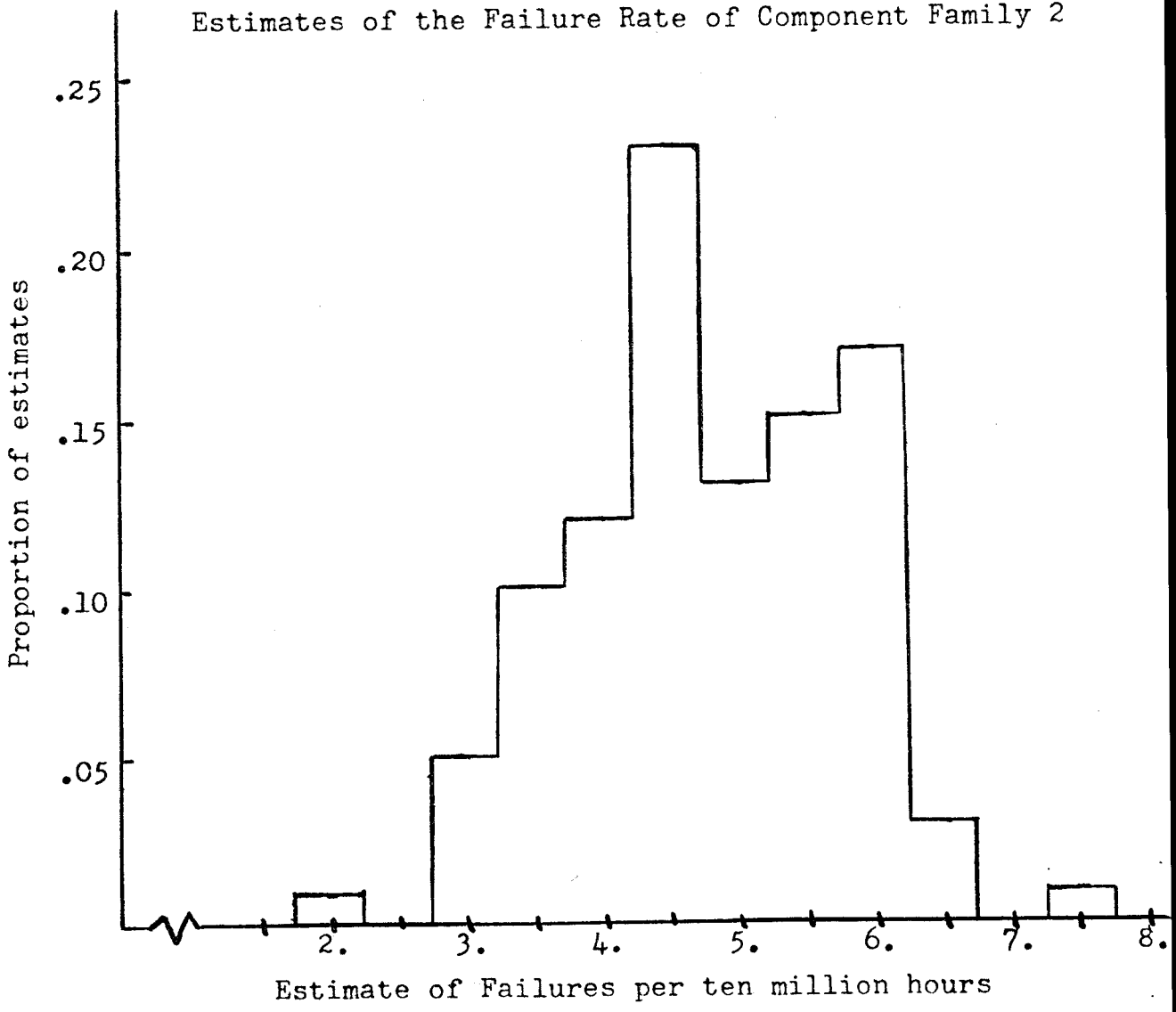
Figure 1



True Value: 10.0 failures/10 million hours
Average of Estimates: 10.166 failures/10 million hours

Figure 2

Distribution of the One Hundred Monte Carlo Estimates of the Failure Rate of Component Family 2

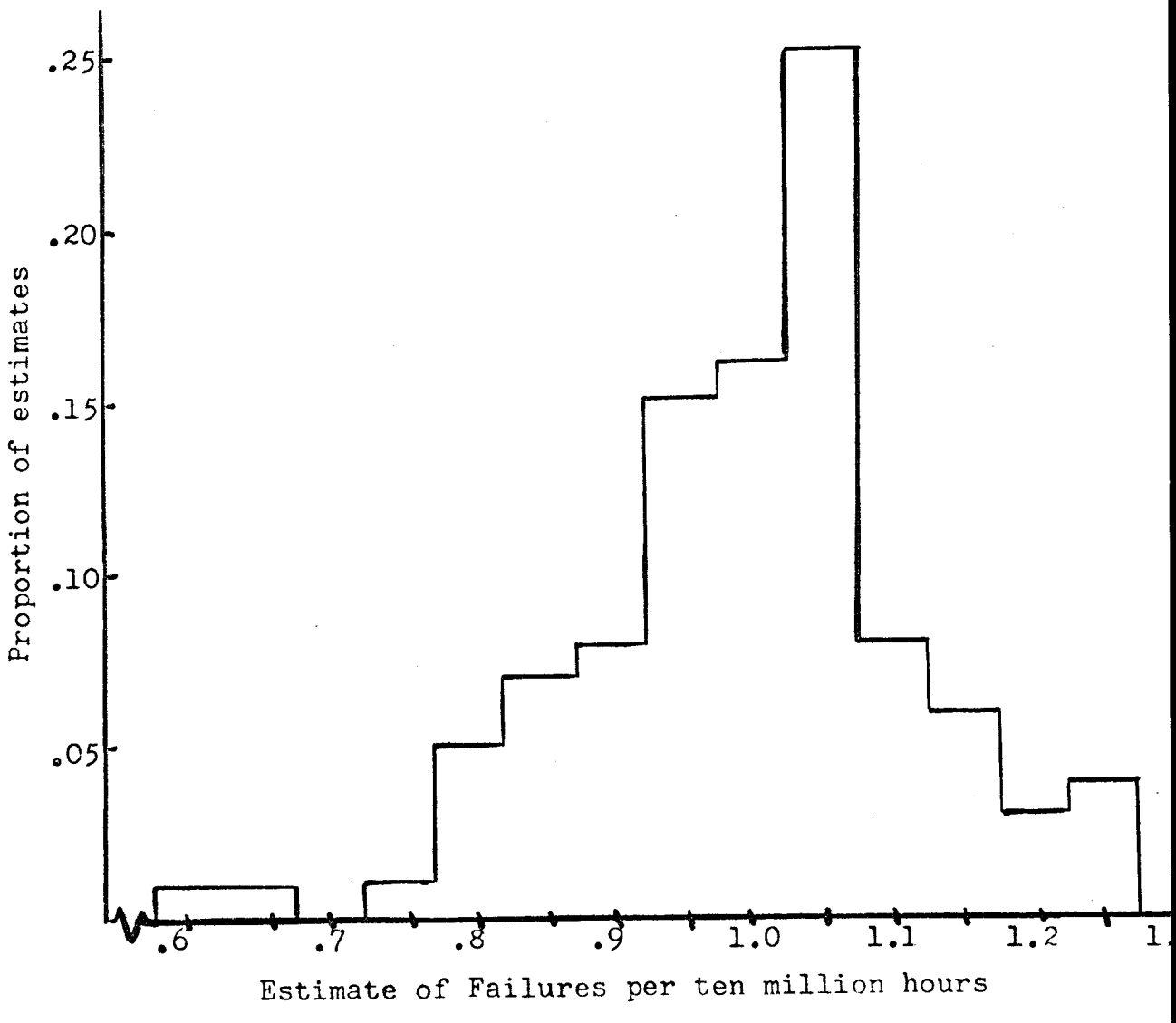


True Value: 5.0 failures/10 million hours

Average of Estimates: 4.830 failures/10 million hours

Figure 3

Distribution of the One Hundred Monte Carlo
Estimates of the Failure Rate of Component Family 3



True Value: 1.0 failures/10 million hours
Average of Estimates: .997 failures/10 million hours

The number of confidence intervals for each family containing the true parameter value indicates that the true level of confidence may be less than ninety-five percent (Table 1). Further Monte Carlo studies with larger samples would allow us to make stronger inferences regarding confidence levels, as the differences between the observed levels of confidence and ninety-five percent may be the result of sampling error.

Since the predicted unit failure rates are linear combinations of the estimates of component family failure rates, it can be expected that predicted unit failure rates will be closer to normal distributions. Evidence of this can be seen from the approximate ninety-five percent confidence intervals for the unit failure rates based on the predicted unit failures. Although still below ninety-five percent, the level of confidence appears to be greater than that for the confidence intervals for component family failure rates. This property will be advantageous when the reliability of a new unit is predicted.

2.3.2 Applications

Information regarding failure rates of electronic units in the field was not available when the computing work was done. As a result, at this time, there is no indication of how well the model fits actual conditions. If unit failure rate data is to be used to estimate component family failure rates, accurate methods of collecting data from users of electronic units must be developed and users must be convinced of the need of accurate data. As the quality

TABLE 1

PERCENTAGE OF APPROXIMATE 95% CONFIDENCE INTERVALS FOR FAILURE RATES CONTAINING THE TRUE PARAMETER VALUE

COMPONENT FAMILIES

COMPONENT FAMILY	PERCENTAGE CONTAINING TRUE VALUE
1	92%
2	93%
3	93%

CONFIDENCE INTERVALS BASED ON PREDICTED UNIT FAILURE RATES

UNIT	PERCENTAGE CONTAINING TRUE VALUE
1	93%
2	95%
3	95%
4	96%
5	95%
6	95%
7	92%
8	92%
9	96%
10	95%

on incoming information improves, the estimates of component family failure rates should improve.

The method of maximum likelihood was applied to a single set of randomly generated approximately Poisson unit failures for one period of time. Ten distinct units made from components of three distinct families were considered. The test of fit statistic calculated from the predicted unit failure rates and observed failure rates was 6.15. This is less than 14.1, the five percent critical value for the Chi-squared distribution on seven degrees of freedom, indicating that this set of data fits the maximum likelihood model. Different starting values were tried for the iterative scheme. It was found that, for this set of generated failure rate data, the scheme converged quickly for starting values between 10^{-24} and 10^{5} failures per ten million hours (Figure 4 and Figure 5). For this set of data the true component family failure rates per ten million hours were ten for the first family, five for the second, and one for the third. This wide range of values for which the method will converge will allow users of the method to proceed without having prior knowledge of approximate failure rates of the components. Future studies of this method, involving randomly generated unit failure rate data, should include more than one period of time.

A lack of sufficient unit failure rate data requires reliability engineers to group components into families. Care must be taken to insure that components are grouped into families of components having the same failure rates. Otherwise, predicted unit

Figure 4

```
N =    0
          ALPHA( 1) =   0.10000003E-23
          ALPHA( 2) =   0.10000003E-23
          ALPHA( 3) =   0.10000003E-23

N =    1
          ALPHA( 1) =   0.11217773E+02
          ALPHA( 2) =   0.45475035E+01
          ALPHA( 3) =   0.89707947E+00

N =    2
          ALPHA( 1) =   0.11386892E+02
          ALPHA( 2) =   0.46041603E+01
          ALPHA( 3) =   0.88496155E+00

N =    3
          ALPHA( 1) =   0.11390028E+02
          ALPHA( 2) =   0.46050749E+01
          ALPHA( 3) =   0.88474065E+00

N =    4
          ALPHA( 1) =   0.11390099E+02
          ALPHA( 2) =   0.46050930E+01
          ALPHA( 3) =   0.88473552E+00

N =    5
          ALPHA( 1) =   0.11390092E+02
          ALPHA( 2) =   0.46050930E+01
          ALPHA( 3) =   0.88473612E+00

N =    6
          ALPHA( 1) =   0.11390099E+02
          ALPHA( 2) =   0.46050949E+01
          ALPHA( 3) =   0.88473552E+00

N =    7
          ALPHA( 1) =   0.11390090E+02
          ALPHA( 2) =   0.46050911E+01
          ALPHA( 3) =   0.88473624E+00

N =    8
          ALPHA( 1) =   0.11390086E+02
          ALPHA( 2) =   0.46050873E+01
          ALPHA( 3) =   0.88473666E+00
```

Sequence of Converging Component Family Failure Rate Estimates using 10^{-24} as Starting Values

Figure 5

```
N =    0
          ALPHA( 1) =   0.10000000E+06
          ALPHA( 2) =   0.10000000E+06
          ALPHA( 3) =   0.10000000E+06

N =    1
          ALPHA( 1) =   0.16000000E+02
          ALPHA( 2) =   0.73750000E+01
          ALPHA( 3) =   0.10000000E+01

N =    2
          ALPHA( 1) =   0.11416687E+02
          ALPHA( 2) =   0.46204128E+01
          ALPHA( 3) =   0.88267040E+00

N =    3
          ALPHA( 1) =   0.11390633E+02
          ALPHA( 2) =   0.46053104E+01
          ALPHA( 3) =   0.88469625E+00

N =    4
          ALPHA( 1) =   0.11390099E+02
          ALPHA( 2) =   0.46050987E+01
          ALPHA( 3) =   0.88473541E+00

N =    5
          ALPHA( 1) =   0.11390092E+02
          ALPHA( 2) =   0.46050930E+01
          ALPHA( 3) =   0.88473600E+00

N =    6
          ALPHA( 1) =   0.11390099E+02
          ALPHA( 2) =   0.46050949E+01
          ALPHA( 3) =   0.88473541E+00

N =    7
          ALPHA( 1) =   0.11390090E+02
          ALPHA( 2) =   0.46050911E+01
          ALPHA( 3) =   0.88473612E+00

N =    8
          ALPHA( 1) =   0.11390087E+02
          ALPHA( 2) =   0.46050911E+01
          ALPHA( 3) =   0.88473642E+00
```

Sequence of Converging Component Family Failure Rate Estimates using 10^5 as Starting Values

failure rates are likely to be poor estimates of the true unit failure rates. Results from accelerated tests and from tests in other fields may be of use in grouping components into families.

CHAPTER 3

YIELD OF COMPONENTS FROM THE YIELD OF UNITS

3.1 Introduction:

Faulty components, damage due to handling during production, and failures expected during normal use contribute to newly produced electronic components failing test procedures. A method of estimating the component family yields from the observed yields of units, the total number of each type of unit produced, and the component makeup of the observed units would enable reliability engineers to predict production yields of newly designed units.

The effect of the production process on one particular component is assumed to be independent of the effect on any other component. Hence, the number of acceptable components of a particular type within a particular unit after production will be a binomial observation. Let X_{ij} be a binomial random variable denoting the number of acceptable components of type j in a unit of type i, that is:

$$P[X_{ij} = x_{ij}] = \binom{q_{ij}}{x_{ij}} A_j^{x_{ij}} (1-A_j)^{q_{ij}-x_{ij}}$$

$$(x_{ij}=0,\ldots,q_{ij};\ j=1,\ldots M;\ i=1,\ldots,N),$$

where q_{ij} is the number of components of type j being acceptable after production, N is the number of distinct types of units to be

used in the estimation procedure, and M is the number of distinct component families found in the N observed units. The probability of a unit of type i being accepted can be expressed as a joint binomial probability, that is:

$$P[X_{i1} = q_{i1},\ldots,X_{iM} = q_{iM}] = \prod_{j=1}^{M} A_j^{q_{ij}} \qquad (i=1,\ldots,N).$$

Denote:

$$p_i = \prod_{j=1}^{M} A_j^{q_{ij}} \qquad (i=1,\ldots,N).$$

It follows from the independence of the acceptability of components, that units are independently produced. Thus, the number of acceptable units of type i produced during a production period will also be binomial. Denoting by the random variable Y_i, the number of acceptable units produced during a production period, we get:

$$P[Y_i=y_i] = \binom{n_i}{y_i} p_i^{y_i} (1-p_i)^{n_i-y_i}$$

$$= \binom{n_i}{y_i} \left[\prod_{j=1}^{M} A_j^{q_{ij}}\right]^{y_i} \left(1-\left[\prod_{j=1}^{M} A_j^{q_{ij}}\right]\right)^{n_i-y_i}$$

$$(0 \leq y_i \leq n_i,\ i=1,\ldots,N)$$

It follows that:

$$E\left[\frac{Y_i}{n_i}\right] = p_i \qquad (i=1,\ldots,N),$$

and:

$$\mathrm{Var}\left[\frac{Y_i}{n_i}\right] = \frac{p_i(1-p_i)}{n_i} \qquad (i=1,\ldots,N).$$

Then using the normal approximation to the binomial distribution, an approximate $100(1-\varepsilon)\%$ confidence interval for a unit of type i is:

$$\frac{y_i}{n_i} \pm z_{\varepsilon/2} \sqrt{\frac{\frac{y_i}{n_i}\left(1-\frac{y_i}{n_i}\right)}{n_i}} \qquad (i=1,\ldots,N).$$

3.2 Method of Maximum Likelihood

Consider N distinct types of electronic units produced using components from M distinct component families. Using the distribution of acceptable components produced, the likelihood equation can be set up:

$$L(A_1,\ldots,A_M|y_1,\ldots,y_N) = \prod_{i=1}^{N} \binom{n_i}{y_i} \left[\prod_{j=1}^{M} A_j^{q_{ij}}\right]^{y_i} \left(1-\left[\prod_{j=1}^{M} A_j^{q_{ij}}\right]\right)^{n_i-y_i}$$

Maximizing the likelihood is equivalent to maximizing the logarithm of the likelihood:

$$\log L(A_1,\ldots,A_M|y_1,\ldots,y_N)$$
$$= \sum_{i=1}^{N} \left\{\log\binom{n_i}{y_i} + y_i \sum_{j=1}^{M} q_{ij} \log A_j + (n_i-y_i) \log\left(1-\left[\prod_{j=1}^{M} A_j^{q_{ij}}\right]\right)\right\}$$

To maximize log L, the derivatives with respect to A_s $(s=1,\ldots,M)$ are found and set equal to zero.

$$\frac{\partial \log L}{\partial A_s} = \sum_{i=1}^{N} \frac{q_{is}\left(y_i - n_i\left[\prod_{j=1}^{M} A_j^{q_{ij}}\right]\right)}{\left(A_s - A_s\left[\prod_{j=1}^{M} A_j^{q_{ij}}\right]\right)} \qquad (s=1,\ldots,M).$$

Set the derivatives equal to zero and solve for $\hat{A}_s$ $(s=1,\ldots,M)$ where $\hat{A}_s$ is the estimated yield for a component of the s^{th} component family:

$$\sum_{i=1}^{N} \frac{q_{is}(y_i - n_i[\prod_{j=1}^{M} \hat{A}_j^{q_{ij}}])}{(\hat{A}_s - \hat{A}_s[\prod_{j=1}^{M} \hat{A}_j^{q_{ij}}])} = 0 \qquad (s=1,\ldots,M).$$

The simplex method developed by Nelder and Mead (Appendix 2) is used to solve for the maximum likelihood estimates of the component family yields. This method proved to be preferable to the method of of scoring. (See page 37.)

The estimated information matrix is used to estimate confidence intervals:

$$\frac{\partial^2 \log L}{\partial A_t \, \partial A_s} = \sum_{i=1}^{N} \frac{q_{is}\, q_{it}[\prod_{j=1}^{M} A_j^{q_{ij}}](y_i - n_i)}{A_s\, A_t(1-[\prod_{j=1}^{M} A_j^{q_{ij}}])^2} \qquad (s=1,\ldots,M;\ t=1,\ldots,M),$$

$$E\left[\frac{-\partial^2 \log L}{\partial A_t \, \partial A_s}\right] = \sum_{i=1}^{N} \frac{n_i\, q_{is}\, q_{it}[\prod_{j=1}^{M} A_j^{q_{ij}}]}{A_s\, A_t(1-[\prod_{j=1}^{M} A_j^{q_{ij}}])}$$

$$= \mathcal{I}(s,t) \qquad (s=1,\ldots,M;\ t=1,\ldots,M),$$

$$\hat{\mathcal{I}}(s,t) = \sum_{i=1}^{N} \frac{n_i\, q_{is}\, q_{it}[\prod_{j=1}^{M} \hat{A}_j^{q_{ij}}]}{A_s\, \hat{A}_t(1-[\prod_{j=1}^{M} \hat{A}_j^{q_{ij}}])} \qquad (s=1,\ldots,M;\ t=1,\ldots,M).$$

Then an approximate $100(1-\varepsilon)\%$ confidence intervals for the s^{th} component family yield is

$$\hat{A}_s \pm z_{\varepsilon/2} \sqrt{\hat{\mathcal{I}}^{-1}(s,s)} \qquad (s=1,\ldots,M),$$

where $z_{\varepsilon/2}$ is the $100\ \varepsilon/2\%$ critical value for the normal distribution and $\hat{\mathcal{I}}^{-1}(s,s)$ is the (s,s) entry of the inverse of the estimated information matrix.

Yields of units can now be predicted using the estimated component family yields:

$$\hat{p}_0 = \prod_{j}^{M} \hat{A}_j^{q_{0j}} \quad ,$$

where $\hat{p}_0$ is the predicted yield of any unit made of components from the M component families for which we estimated yields and q_{0j} is the number q_{0j} is the number of components of type j in that unit. The predicted yields of the N units used in the estimation procedure can be compared with the confidence intervals for unit yields based on the observed values.

An approximate test of fit can be performed using the normal approximation to the binomial distribution. The statistic

$$\sum_{i=1}^{N} \frac{(y_i - n_i \hat{p}_i)^2}{n_i \hat{p}_i(1-\hat{p}_i)}$$ is asymptotically Chi squared on N-M degrees of freedom.

3.3 Method of Weighted Least Squares

Consider N distinct types of electronic units produced using components from M distinct component families. Take the logarithm of the yield of a unit of type i:

$$\log p_i = \sum_{j=1}^{M} q_{ij} \log A_j \qquad (i=1,\ldots,N).$$

Let Y_i be the random variable for the acceptable number of units produced if a total of n_i units of type i are produced. Then, since the distribution of acceptable units produced is assumed to be binomial the exact mean and variance of Y_i/n_i are known. The exact

variance of log (Y_i/n_i), however, is unknown but can be approximated.

Let:

$$b\left(\frac{Y_i}{n_i}\right) = \log\,(Y_i/n_i) \qquad (i=1,\ldots,N),$$

$$b'\left(\frac{Y_i}{n_i}\right) = \frac{1}{Y_i/n_i} \qquad (i=1,\ldots,N).$$

By Taylor's Theorem:

$$b\left(\frac{Y_i}{n_i}\right) \simeq b(p_i) + \left(\frac{Y_i}{n_i} - p_i\right) b'(p_i),$$

where y_i is an observed value of Y_i.

Then:

$$\mathrm{Var}\left[b\left(\frac{Y_i}{n_i}\right)\right] \simeq \{b'(p_i)\}^2\,\mathrm{Var}\left[\frac{Y_i}{n_i}\right]$$

$$= \left(\frac{1}{p_i}\right)^2 \frac{p_i(1-p_i)}{n_i}$$

$$\simeq \frac{n_i - y_i}{n_i\, y_i}$$

$$= w_i^{\,2}\,, \text{ say} \qquad (i=1,\ldots,N).$$

The method of weighted least squares can be applied:

$$\hat{A} = (Q_*'Q_*)^{-1}\, Q_*'\, y_*$$

where:

$$\hat{A} = \begin{bmatrix} \log \hat{A}_1 \\ \vdots \\ \log \hat{A}_M \end{bmatrix}, \quad y_* = \begin{bmatrix} \dfrac{\log y_1}{w_1} \\ \vdots \\ \dfrac{\log y_N}{w_N} \end{bmatrix}, \quad Q_* = \left[q_{ij}/w_i\right]_{N\times M}.$$

Approximate 100(1-ε)% confidence intervals for the logarithm of the component family yields are:

$$\log \hat{A}_s \pm t_{\varepsilon/2}(N-M) \sqrt{(Q_*'Q_*)^{-1}_{ss} \hat{\sigma}^2} \qquad (s=1,\ldots,M),$$

where $t_{\varepsilon/2}(N-M)$ is the $100\ \varepsilon/2\%$ critical value for the student's t distribution on N-M degrees of freedom, $(Q_*'Q_*)^{-1}$ is the (s,s) entry of the matrix $(Q_*'Q_*)^{-1}$ and $\hat{\sigma}^2$ is an estimate of the residual variance:

$$\hat{\sigma}^2 = \frac{(y_*-Q_*\hat{A})'(y_*-Q_*\hat{A})}{N-M}$$

$$= \frac{y_*'y_* - \hat{A}\,Q_*'}{N-M}$$

By taking the exponent of the bounds of the confidence intervals, approximate $100(1-\varepsilon)\%$ confidence intervals for the estimates of the component family yields can be found:

$$\hat{A}_s \exp(\pm t_{\varepsilon/2}(N-M) \sqrt{(Q_*'\,Q_*)^{-1}_{s,s}\hat{\sigma}^2}) \qquad (s=1,\ldots,M).$$

Yields of units can now be predicted using the estimates of component family yields:

$$\hat{p}_0 = \prod_{j=1}^{M} \hat{A}_j^{q_{0j}}$$

where $\hat{p}_0$ is the predicted yield of any unit composed of components for which estimates have been made and q_{0j} is the number of components of type j in that unit. Since

$$\log \hat{p}_0 = \sum_{j=1}^{M} q_{0j} \log \hat{A}_j ,$$

it follows that:

$$\mathrm{Var}(\log \hat{p}_0) = \mathrm{Var}\left(\sum_{j=1}^{M} q_{0j} \log \hat{A}_j\right)$$

$$= \sum_{k=1}^{M} \sum_{j=1}^{M} q_{0j}\, q_{0k}\, \mathrm{Cov}(\log \hat{A}_j, \log \hat{A}_k)$$

$$= \sum_{k=1}^{M} \sum_{j=1}^{M} q_{0j}\, q_{0k}\, \hat{\sigma}^2\, (Q_*'\, Q_*)^{-1}_{j,k} \cdot$$

Then an approximate $100(1-\varepsilon)\%$ confidence interval for the predicted yield will be:

$$\hat{p}_0 \cdot \exp\left(\pm z_{\varepsilon/2} \sqrt{\sum_{k=1}^{M} \sum_{j=1}^{M} q_{0j}\, q_{0k}\, \hat{\sigma}^2\, (Q_*'\, Q_*)^{-1}_{j,k}}\right)$$

The approximate $100(1-\varepsilon)\%$ confidence intervals for the unit yields based on the predicted yields can be compared with these based on the observed yields.

An approximate test of fit can be performed using the normal approximation to the binomial distribution.

The statistic $\sum_{i=1}^{N} \frac{(y_i - n_i \hat{p}_i)^2}{n_i \hat{p}_i (1-\hat{p}_i)}$ is asymptotically Chi squared on N-M degrees of freedom.

3.4 Discussion and Recommendations

3.4.1 Monte Carlo Study:

Ten distinct artificial units, each made of components from three distinct families with known yields, were simulated. The number of each type of unit produced during a production period was chosen to represent possible actual conditions. One hundred samples of the number of accepted units of each of the ten types were randomly generated using the normal approximation to the binomial distribution. If Y_i is binomial with mean $n_i\, p_i$ and variance

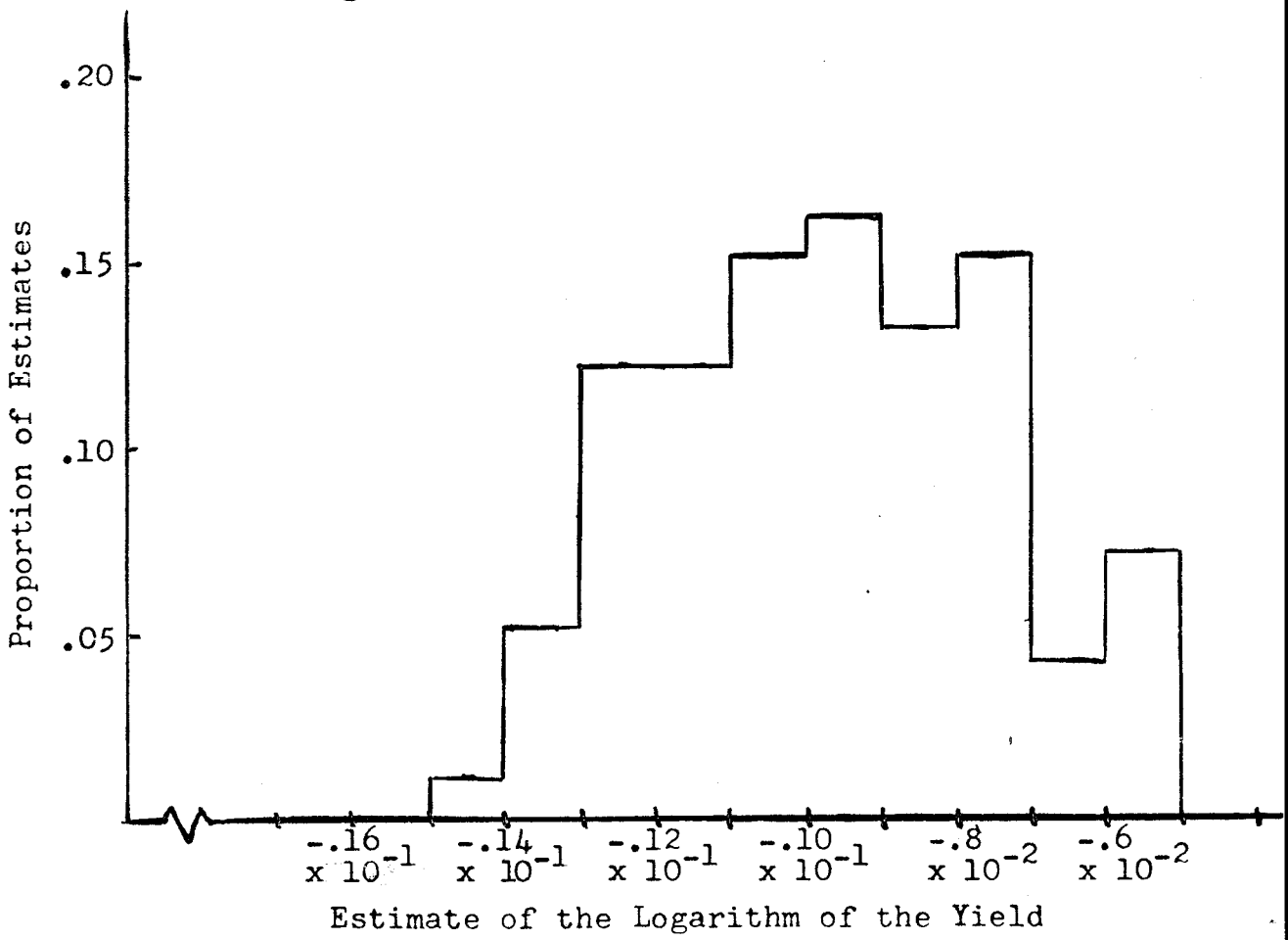
$n_i\ p_i(1-p_i)$, then Y_i is approximately normal with mean $n_i\ p_i$ and variance $n_i\ p_i(1-p_i)$. The method of weighted least squares was then applied to the one hundred samples.

Approximate ninety-five percent confidence intervals for each of the logarithms of the component family yields for the one hundred samples were calculated using the normal approximation commonly used in least squares analyses. The estimates of the logarithms of the component family yields made from the generated unit yield data appear to be approximately normal (Figures 6,7 and 8). Under actual conditions it can be expected that the same will hold true. The number of confidence intervals for each family containing the true parameter value indicates that the true level of confidence may be greater than ninety-five percent (Table 2). One explanation for this may be that the weights used to adjust the component makeup matrix Q were approximate. Another possible explanation may be the non-normality of the linearized unit yield data.

Future studies could try using the information matrix for the linearized problem as the variance-covariance matrix. Confidence intervals using the inverse of this information matrix would not then depend on the adjusted component makeup matrix. To determine this information matrix, let $A_1,\ldots,A_M$ be the M component family yields, L be the likelihood function for $A_1,\ldots,A_M$ with N observed unit yields, and q_{ij} be the number of components of type j in a unit of type i. It can be shown that:

Figure 6

Distribution of the One Hundred Monte Carlo
Estimates of the Logarithm of the Yield of Component Family 1

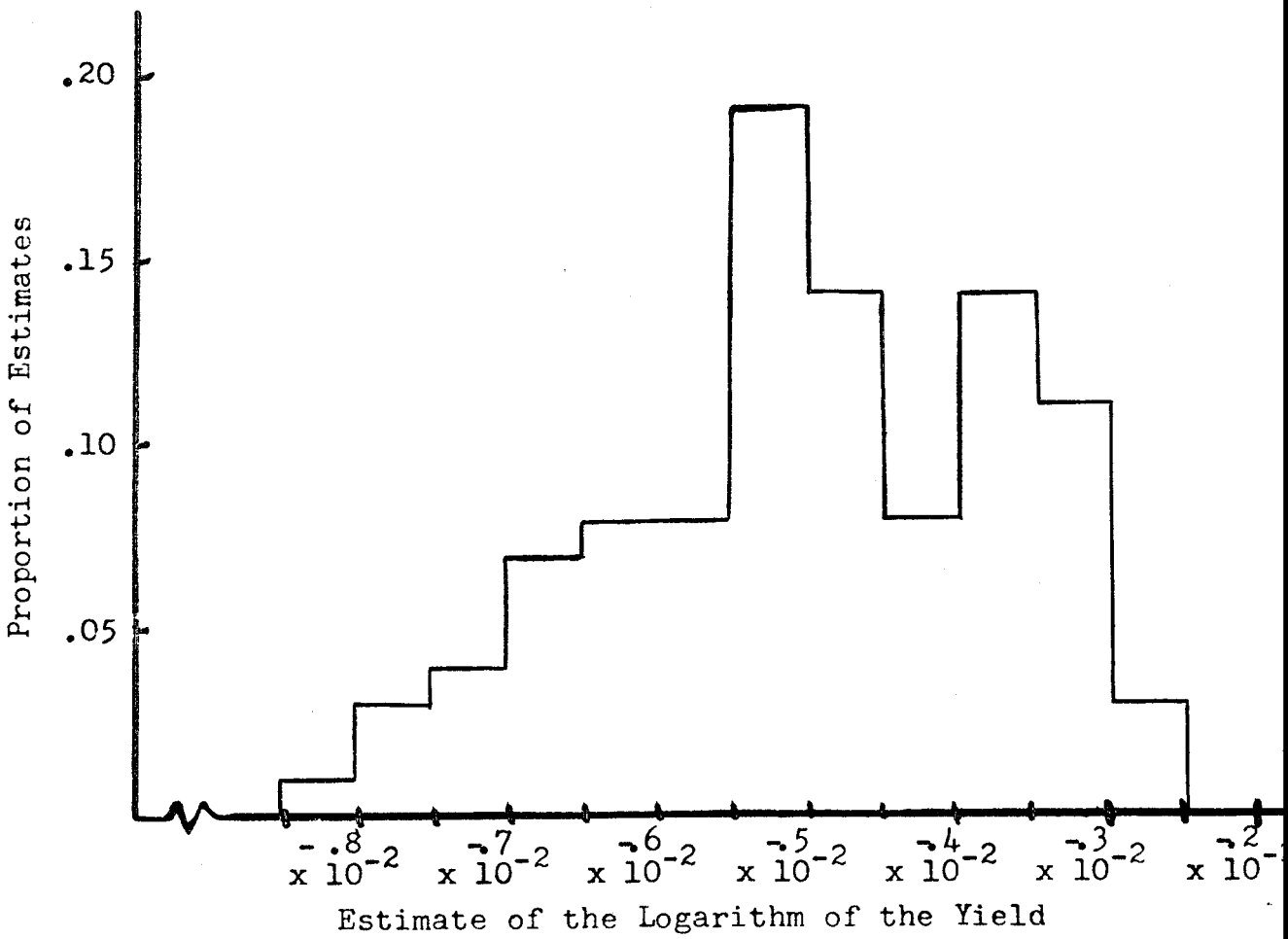


True Value: $\log(.99) = -.100 \times 10^{-1}$

Average of Estimates: $-.97283 \times 10^{-2}$
$= \log(.99032)$

Figure 7

Distribution of the One Hundred Monte Carlo
Estimates of the Logarithm of the Yield of Component Family 2



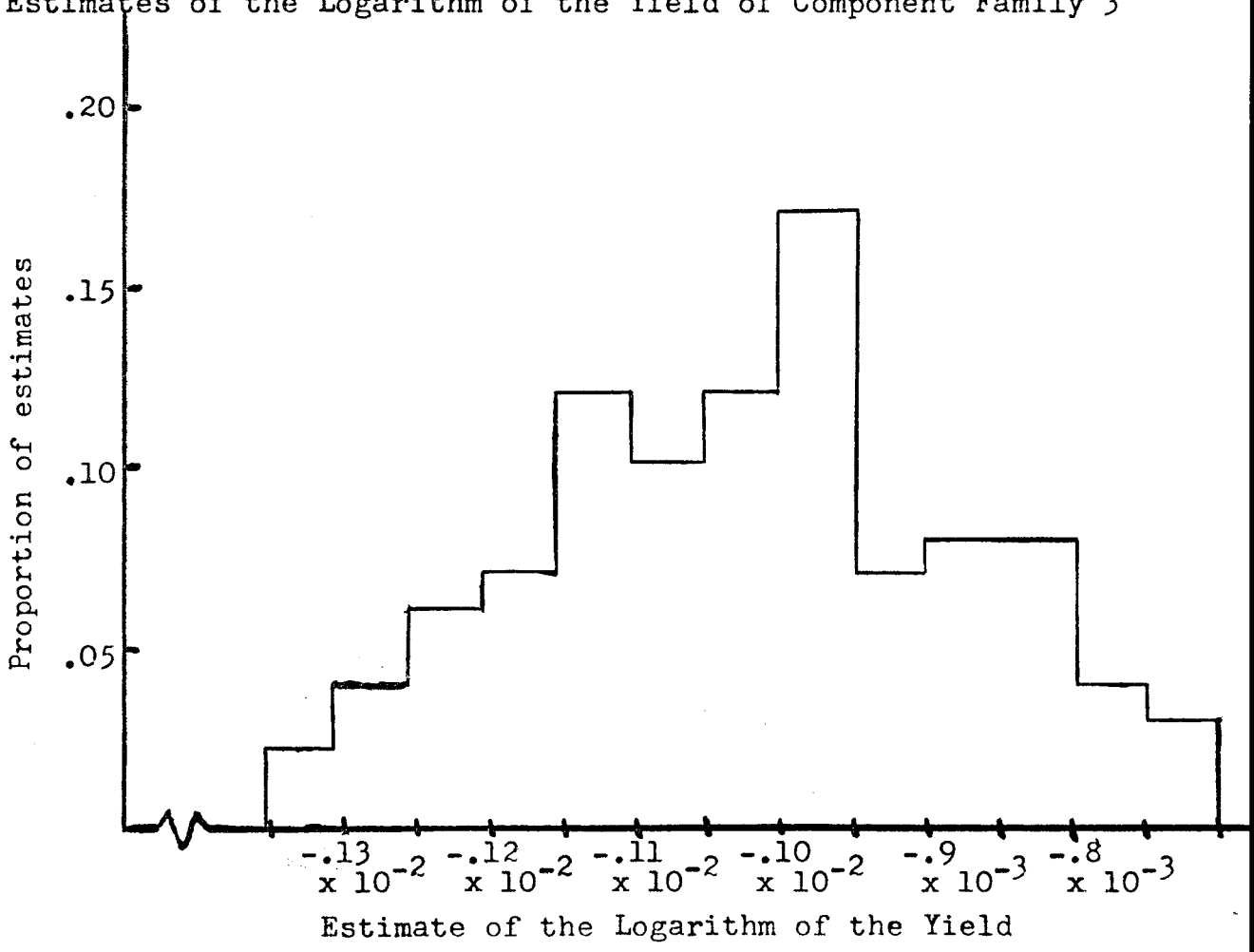
True Value: $\log(.995) = -.50125 \times 10^{-2}$

Average of Estimates: $-.50201 \times 10^{-2}$

$= \log(.99499)$

Figure 8

Distribution of the One Hundred Monte Carlo Estimates of the Logarithm of the Yield of Component Family 3



True Value: $\log(.999) = -.10005 \times 10^{-2}$

Average of Estimates: $-.10156 \times 10^{-2}$

$= \log(.99898$

TABLE 2

PERCENTAGE OF APPROXIMATE 95% CONFIDENCE INTERVALS FOR YIELDS CONTAINING THE TRUE PARAMETER VALUE

COMPONENT FAMILIES

COMPONENT FAMILY	PERCENTAGE CONTAINING TRUE VALUE
1	97%
2	99%
3	97%

UNITS (CONFIDENCE INTERVALS BASED ON PREDICTED YIELDS)

UNIT	PERCENTAGE CONTAINING TRUE VALUE
1	96%
2	91%
3	94%
4	91%
5	90%
6	94%
7	89%
8	94%
9	92%
10	96%

$$\frac{\partial^2 \log L}{\partial \log A_t \, \partial \log A_s} = \frac{1}{A_s A_t} \frac{\partial^2 \log L}{\partial A_t \, \partial A_s} \qquad (s=1,\ldots,M;\ t=1,\ldots,M).$$

Then using the results of Section 3.2:

$$\frac{\partial^2 \log L}{\partial \log A_i \, \partial \text{ Log } A_s} = \sum_{i=1}^{N} \frac{q_{is}\, q_{it} [\prod_{j=1}^{M} A_j^{q_{ij}}](y_i - n_i)}{A_s^2 A_t^2 (1-[\prod_{j=1}^{M} A_j^{q_{ij}}])^2}$$

$$(s=1,\ldots,M;\ t=1,\ldots,M).$$

It can then be shown that:

$$E\left[-\frac{\partial^2 \log L}{\partial \log A_t \, \partial \log A_s}\right] = \sum_{i=1}^{N} \frac{n_i\, q_{is}\, q_{it} [\prod_{j=1}^{M} A_j^{q_{ij}}]}{A_s^2 A_t^2 (1-[\prod_{j=1}^{M} A_j^{q_{ij}}]}$$

$$(s=1,\ldots,M;\ t=1,\ldots,M).$$

A likelihood ratio test was used to test equality of the estimated variance covariance matrix of the one hundred sets of component family yields and the inverse of the true information matrix (T.W. Anderson, 1958, Section 10.8). The calculated statistic was 304.44. The five percent Chi squared critical value on six degrees of freedom is 12.6, meaning that the hypothesis of equality of the two matrices is rejected at the five percent level. Although shown to be unequal, it appears that the inverse of the information matrix is an adequate approximation to the variance-covariance matrix for most purposes (Compare Table 3 and Table 4). The use of the inverse of the estimated information matrix in calculating confidence intervals should not greatly alter the level of confidence from ninety-five percent.

TABLE 3

ESTIMATED VARIANCE-COVARIANCE MATRIX FOR THE ONE HUNDRED SAMPLES OF LOGARITHMS OF THE COMPONENT FAMILY YIELDS

4.9178×10^{-6}	5.6114×10^{-7}	-2.9472×10^{-7}
5.6114×10^{-7}	1.6996×10^{-6}	-5.2049×10^{-8}
-2.9472×10^{-7}	-5.2049×10^{-8}	2.4009×10^{-8}

TABLE 4

INVERSE OF THE TRUE INFORMATION MATRIX FOR THE LOGARITHMS OF THE COMPONENT FAMILY YIELDS

4.4645×10^{-6}	2.9749×10^{-7}	-2.8201×10^{-7}
2.9749×10^{-7}	1.7344×10^{-6}	-4.7913×10^{-8}
-2.8301×10^{-7}	-4.7913×10^{-8}	1.9383×10^{-8}

Since the predicted logarithms of the unit yields are linear combinations of the estimates of the logarithms of the component family yields, it can be expected that the predicted logarithms of the unit yields will be better approximations to normal distributions. A poor estimate of the yield of components of the second family may explain why the level of confidence for some of the intervals based on predicted yields appear significantly different from ninety-five percent while others are relatively close. It appears that where components of the second family in a yield estimate, the level of confidence for the confidence interval for that estimate is further from ninety-five percent. (Compare Table 2 and Table 5). Having used only one hundred sets of observations, sampling error may explain the difference between the observed level of confidence and ninety-five percent. Further Monte Carlo studies with larger samples may be of use in making conclusions.

3.4.2 Applications

The method of predicting unit yields presently used treats an insertion as a component. However, an insertion is not a component but a process. An insertion may be a success or a failure, but, when units are tested after production, it is the components which are acceptable or unacceptable, not the insertions. A component may be inserted into a unit manually or by machine. If the probability of one type of insertion being successful is different from the probability of the other type being acceptable, then the yield of a

TABLE 5

COMPONENT FAMILY STRUCTURE OF THE ARTIFICIAL UNITS USED FOR THE MONTE CARLO STUDY OF THE WEIGHTED LEAST SQUARES METHOD FOR COMPONENT YIELDS

		COMPONENT FAMILY		
		1	2	3
Unit	1	27	3	450
	2	25	10	350
	3	31	0	375
	4	16	11	400
	5	10	25	200
	6	35	0	425
	7	29	29	325
	8	21	0	400
	9	21	9	428
	10	17	25	216

particular component family inserted manually should be estimated separately from the yield of components of that family inserted by machine.

Both the method of weighted least squares and the method of maximum likelihood were applied to a single set of randomly generated unit yields for those ten artificial units used in the Monte Carlo study of section 3.4.1. The results indicate that both methods work for the randomly generated approximately binomial data. However, in the case of the maximum likelihood method, numerical difficulties were encountered.

Using the weighted least squares estimates, the test of fit statistic for the randomly generated data was 2.23. The Chi squared critical value on seven degrees of freedom is 14.1. The test of fit statistic was also calculated for two of the samples used in the Monte Carlo study and were also found to be acceptable at the ninety-five percent level. It then appears that the weighted least squares method is satisfactory for approximately binomial observations of acceptable components. However, when the weighted least squares method was applied to actual production yields, the fit was not acceptable. Two sets of actual production data were used for which the units were made from five distinct families. The test of fit statistic for the one set of data was 10.13 which is greater than 7.81, the Chi squared five percent critical value on three degrees of freedom. For the other set of data, the statistic was 396.29 which is greater than 27.6, the Chi squared five percent critical

value on seventeen degrees of freedom. Although sampling error may be responsible for the very poor fit of the model to the second set of actual production yield data, a number of other factors may be responsible.

The present system of predicting yields of new units groups components into families which may not have the same yields. For example, resistors of all types are considered to be one family. However, it may be that resistors of different specifications and construction are affected differently during the production process. The possibility that the yield of a component inserted manually may differ from the yield of a component inserted by machine was not considered. Tests should be made to insure that all members of a family have the same yield. Otherwise, predicted unit yields are likely to be poor estimates of the true unit yields. Those component families used in the estimation of failure rates may not be appropriate for use in estimating yields.

A number of factors in the production process itself may cause the observed number of acceptable components to deviate from the binomial distribution. Tests may vary from one type of unit to another. If units are tested for any length of time, then those Poisson failures which can be expected during normal operation of a unit will noticeably affect the distribution of the number of units passing production tests. The yield of the first few units produced may be less than that for units produced after any initial production problems have been corrected. Human factors, such as one worker being distracted by another, may cause units to become unacceptable after production. The production process should

be checked to find out how these factors affect production. If units are produced in more than one location, checks should be made to insure that the yields are independent of location before results from different locations are combined to make yield estimates.

The method of maximum likelihood was initially tried using the method of scoring (Appendix 1) to maximize the likelihood. The program PREDMLE, designed to maximize the likelihood using the method of scoring was tested (Appendix 4) and the logic was found to be correct. Using the true unit yields as input and using the true component family yields as starting values, the iterative scheme converged after three steps. However, using other starting values close to the true component family yields, the estimates did not appear to be converging after seventy-five iterations (Table 6). This method was also tried with actual unit yield data from a production line. As anticipated, the sequence of component family yield estimates diverged. Weighted least squares estimates of the component family yields were used as starting values for the iterative procedure. However, after one hundred and fifty iterations, the component family yield estimates did not appear to be converging. The method of scoring was then discarded as a practical method of finding maximum likelihood estimates of component family yields.

The simplex optimization method of Nelder and Mead was then used to find those component family yield estimates which maximize the yield. This is the method described in section 3.2. Weighted least square estimates of the component family yields were used as

TABLE 6

ESTIMATED COMPONENT FAMILY YIELDS USING THE METHOD OF SCORING TO MAXIMIZE THE LIKELIHOOD

Component Family	Starting Values	Estimated Yield after 3 Iterations	Expected Value of Estimate
1	.9900	.9900	.9900
2	.9950	.9950	.9950
3	.9990	.9990	.9990

Component Family	Starting Values	Estimated Yield after 3 Iterations	Expected Value of Estimate
1	.9925	.9996	.9900
2	.9950	.9969	.9950
3	.9975	.9982	.9990

one of the vertices of the initial simplex. When the method was applied to the randomly generated set of data, it was found that the likelihood increased as the procedure progressed (Figure 9). Although the method appears to work, the value of the likelihood remained constant after a number of steps. This was a result of the limited number of figures of accuracy kept by the computer. When the method was applied to actual observed yields of units produced, the same observations were made. One possible solution to the restricted number of figures of accuracy may be to use double precision in the computational work. It was also noted that, for both the randomly generated set of data and the actual observations, the maximum likelihood method generally gave a slightly smaller Chi-squared value than did the method of weighted least squares.

As an alternative to the method of maximum likelihood, the method of minimum Chi squared estimation (Kendall and Stuart, Volume 3, 1973, section 19.25) could be tried. This method minimizes the test of fit statistic:

$$\sum_{i=1}^{N} \frac{(n_i \hat{p}_i - y_i)^2}{n_i \hat{p}_i (1-\hat{p}_i)}$$

where N is the number of distinct types of units produced, n_i is the number of units of type i produced, $\hat{p}_i$ is the predicted yield of a unit of type i, and y_i is the observed number of units of type i accepted after production. One possible method of minimizing this statistic is to set equal to zero the derivatives of the statistic with respect to the estimated component family yield estimates. If the range of

Figure 9

Step by Step Results of Application of Simplex Method
of Maximizing the Likelihood For a Randomly Generated Set of Observations

NCOUNT = 1	SHRINK	LIKELIHOOD = -0,370235303 E+04
NCOUNT = 2	EXPANSION	LIKELIHOOD = -0,370235303 E+04
NCOUNT = 3	SHRINK	LIKELIHOOD = -0,369542139 E+04
NCOUNT = 4	REFLECTION	LIKELIHOOD = -0,369542139 E+04
NCOUNT = 5	SHRINK	LIKELIHOOD = -0,369542139 E+04
NCOUNT = 6	SHRINK	LIKELIHOOD = -0,369542139 E+04
NCOUNT = 7	SHRINK	LIKELIHOOD = -0,369542139 E+04
NCOUNT = 8	SHRINK	LIKELIHOOD = -0,369542139 E+04
NCOUNT = 9	REFLECTION	LIKELIHOOD = -0,369542139 E+04
NCOUNT = 10	SHRINK	LIKELIHOOD = -0,369541650 E+04
NCOUNT = 11	EXPANSION	LIKELIHOOD = -0,369541650 E+04
NCOUNT = 12	SHRINK	LIKELIHOOD = -0,369540625 E+04
NCOUNT = 13	REFLECTION	LIKELIHOOD = -0,369540625 E+04
NCOUNT = 14	SHRINK	LIKELIHOOD = -0,369540601 E+04
NCOUNT = 15	EXPANSION	LIKELIHOOD = -0,369540601 E+04
NCOUNT = 16	SHRINK	LIKELIHOOD = -0,369540259 E+04
NCOUNT = 17	REFLECTION	LIKELIHOOD = -0,369540259 E+04
NCOUNT = 18	SHRINK	LIKELIHOOD = -0,369540259 E+04
NCOUNT = 19	EXPANSION	LIKELIHOOD = -0,369540259 E+04
NCOUNT = 20	SHRINK	LIKELIHOOD = -0,369540161 E+04
NCOUNT = 21	EXPANSION	LIKELIHOOD = -0,369540161 E+04
NCOUNT = 22	SHRINK	LIKELIHOOD = -0,369540137 E+04
NCOUNT = 23	SHRINK	LIKELIHOOD = -0,369540137 E+04
NCOUNT = 24	EXPANSION	LIKELIHOOD = -0,369540137 E+04
NCOUNT = 25	REFLECTION	LIKELIHOOD = -0,369540137 E+04
NCOUNT = 26	EXPANSION	LIKELIHOOD = -0,369540137 E+04
NCOUNT = 27	SHRINK	LIKELIHOOD = -0,369540137 E+04
NCOUNT = 28	EXPANSION	LIKELIHOOD = -0,369540137 E+04
NCOUNT = 29	SHRINK	LIKELIHOOD = -0,369540015 E+04
NCOUNT = 30	EXPANSION	LIKELIHOOD = -0,369540015 E+04
NCOUNT = 31	SHRINK	LIKELIHOOD = -0,369539990 E+04
NCOUNT = 32	EXPANSION	LIKELIHOOD = -0,369539990 E+04
NCOUNT = 33	SHRINK	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 34	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 35	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 36	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 37	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 38	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 39	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 40	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 41	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 42	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 43	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 44	REFLECTION	LIKELIHOOD = -0,369539844 E+04
NCOUNT = 45	REFLECTION	LIKELIHOOD = -0,369539844 E+04

convergence is great enough, these equations may be solved using Newton's iterative method (Russell, 1970, Chapter 5). Otherwise the statistic may be minimized using the simplex method of Nelder and Mead. (See Appendix 2). One would expect that the number of figures of accuracy available with the computer is sufficient for making satisfactory estimates of component family yields.

Although the method of weighted least squares may give reasonable estimates of component yields, further work with this problem may be of value. The method of weighted least squares has the advantage of requiring comparatively few calculations. However, we can expect the maximum likelihood estiamtion method and the minimum Chi squared estimation method to fit the data better. It may be that a wide range of starting values can be used for the iterative scheme for the minimum Chi squared method and the sequence of estimates for this method may converge quickly. To use either the maximum likelihood or the minimum Chi squared methods for practical purposes, a method of calculating approximate confidence intervals must be found.

In order to make any of the methods of estimation better for practical use, there are two areas which should be investigated. The yields of the components within each family should be tested for equality. If equality among the components of a family does not exist, the components should be regrouped into families of components of equal yield. The second area to be investigated is the actual production process. One can expect that the closer the observed

number of units accepted after production is to a binomial observation, the better will be the estimates of component family yields. An investigation into these two areas may provide reliability engineers with better estimates of component yields and, hence, better predictions of unit yields.

APPENDIX 1

The Method of Scoring

The method of scoring (C.R. Rao, 1965, Section 5g) is an iterative scheme for finding maximum likelihood estimates of parameters. Define L as the likelihood function for the vector of parameters:

$$\underset{\sim}{\theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_N \end{bmatrix} .$$

Let:

$$S(\underset{\sim}{\theta}) = \begin{bmatrix} \frac{\partial \log L}{\partial \theta_1} \\ \vdots \\ \frac{\partial \log L}{\partial \theta_N} \end{bmatrix} , \quad \mathcal{I}(\underset{\sim}{\theta}) = E\left[-\frac{\partial^2 \log L}{\partial \theta_i \partial \theta_j} \right]_{N \times N}$$

Let $H(\underset{\sim}{\theta})$ be the N by N derivative matrix for $S(\underset{\sim}{\theta})$.

Then:

$$E[-H(\underset{\sim}{\theta})] = \mathcal{I}(\underset{\sim}{\theta}).$$

As a first approximation, one may take:

$$\mathcal{I}(\hat{\underset{\sim}{\theta}}) \simeq -H(\underset{\sim}{\theta}),$$

where $\hat{\underset{\sim}{\theta}}$ is the maximum likelihood estimate of $\underset{\sim}{\theta}$.

Since a first order Taylor's expansion gives:

$$S(\underset{\sim}{\theta}) \simeq S(\hat{\underset{\sim}{\theta}}) + (\underset{\sim}{\theta} - \hat{\underset{\sim}{\theta}})\, H(\hat{\underset{\sim}{\theta}}),$$

and:

$$S(\hat{\underset{\sim}{\theta}}) = 0, \quad \text{and we derive the iterative scheme:}$$

$$\underset{\sim}{\theta}_{\ell+1} = \underset{\sim}{\theta}_{\ell} - H^{-1}(\underset{\sim}{\theta}_{\ell}) \cdot S(\underset{\sim}{\theta}_{\ell}),$$

which will converge to $\hat{\underset{\sim}{\theta}}$ if an initial value $\underset{\sim}{\theta}_0$ sufficiently close to $\hat{\underset{\sim}{\theta}}$ is chosen. Replacing H by its expected value, we get:

$$\underset{\sim}{\theta}_{\ell+1} = \underset{\sim}{\theta}_{\ell} - \mathcal{I}^{-1}(\underset{\sim}{\theta}_{\ell}) \cdot S(\underset{\sim}{\theta}_{\ell}),$$

which is the iterative method of scoring.

APPENDIX 2

The Nelder-Mead Simplex Method:

The Nelder Mead simplex (Kowalik and Osborne, 1968, Section 2.6) is a direct search technique for finding an optimum value. The procedure as described here locates a maximum using a simplex, a set of n+1 points in n dimensions. The simplex is manipulated by reflecting the point at which the function f is the least, or by expanding or contracting the simplex

Let:

(1) x_h be the vertex corresponding to $f(x_h) = \min_i f(x_i)$ $(i=1,\ldots,n+1)$.

(2) x_s be the vertex which corresponds to $f(x_s) = \min_i f(x)$, $i \neq h$.

(3) x_ℓ be the vertex corresponding to $f(x_\ell) = \max_i f(x_i)$, $(i=1,\ldots,n+1)$.

(4) x_0 be the centroid of all x_i, $i \neq h$ and is given by

$$x_0 = \frac{1}{n} \sum_{\substack{i=1 \\ i \neq h}}^{n+1} x_i \ .$$

Now define the three basic operations used:

(1) Reflection, where x_h is replaced by

$$x_r = (1+\alpha)x_0 - \alpha x_h$$

where $\alpha > 0$ and is equal to the ratio of the distance $[x_r x_0]$ to $[x_h x_0]$.

(2) Expansion, where x_r is expanded in the direction along which the function value is expected to increase. The relation

$$x_e = \gamma x_r + (1-\gamma)x_0$$

is used where $\gamma > 1$ is the ratio of the distance $[x_e, x_0]$ to $[x_r, x_0]$.

(3) Contraction, where the simplex is contracted,

$$x_c = \beta\, x_h + (1-\beta)x_0,$$

where $0 < \beta < 1$ and is the ratio of the distance $[x_c, x_0]$ to $[x_h, x_0]$.

The technique proceeds as follows:

(i) From an initial simplex and evaluate the function at each of the n+1 vertices.

(ii) Try a reflection and evaluate the function at the reflected point.

(iii) If $f(x_h) \lesssim f(x_r) \lesssim f(x_\ell)$, replace x_h by x_r and return to step (i).

(iv) If $f(x_r) > f(x_\ell)$ it can be expected that an expansion in the direction $x_r - x_0$ could give us a greater value f or f. Replace x_h by x_r if $f(x_\ell) < f(x_e)$ and return to step (i). Otherwise replace x_h by x_r and return to step (i).

(v) If $f(x_h) < f(x_r) < f(x_s)$ replace x_h by x_r and try a contraction. If $f(x_h) < f(x_c)$, replace x_h by x_c and return to step (i). Otherwise the last simplex is shrunk about the point x_ℓ by the

relation

$$x_i' = \frac{1}{2}(x_i + x_\ell)$$

and return to (i).

APPENDIX 3

Programs for the Component Failure Rate Problem

The computing work for the problem of estimating component family failure rates involved two programs. Both programs are written in FORTRAN and both use the maximum likelihood method of estimating component family failure rates for one period of time. The program FAILMLE is designed for practical use and an external file is used as its source of unit failure rate data. The program MONTE generates one hundred random samples of unit failure rates for which component family failure rates are estimated. The input files for both programs are eighty columns long. The output files have a length of one hundred and thirty-three columns including carriage control.

Input for FAILMLE (see Figure 10) is unformatted and is completely general for any number of units observed and any number of component families. The first row of the input gives N, the number of distinct units observed, and then M, the number of distinct component families for which failure rates can be estimated. The first M columns of the second to (N+1)st rows, inclusive, is the matrix $Q = [q_{ij}]_{N\times M}$ where q_{ij} is the number of components of the j^{th} family in a unit of type i. The (M+1)st entry in the (i+1)st row is the number of units of type i in use during the time period. The (M+2)nd entry in the (i+1)st row is the number of type i unit failures observed during that time period. The (N+2)nd row give the length of the time period in units of ten million hours for which observations were made. The value in the (N+3)rd row is given to the

M component family failure rates as starting values in the iterative procedure.

Input for MONTE is the same as that for FAILMLE except that M is restricted to three. Input for unit failures of one sample is required, but the results of the program are independent of this information. An alteration to the program would allow the omission of unit failures as input. MONTE could be generalized to allow for any number of component families and for any number of generated samples.

The external subroutines INFAIL and LAFAIL are called from both FAILMLE and MONTE. INFAIL determines the estimated information matrix as described in section 2.2 and LAFAIL calculates the predicted failure rates of those units which have been used in the estimation procedure. Three subroutines, SINV,EIGEN, and GAUSS, from the SSP FORTRAN library were used in the computing. SINV, which determines the inverse of a positive definite symmetric matrix, is called from INFAIL. The subroutine EIGEN is used to determine the eigenvectors and eigenvalues of the inverse of the estimated information matrix in the program FAILMLE. GAUSS, a normal random number generator, is used in MONTE to generate unit failures.

The approximate confidence intervals in both FAILMLE and MONTE are calculated at the ninety-five percent confidence level. Alterations in the program could be made which would allow different levels of confidence. FAILMLE could also be generalized to allow

for observations of unit failure rates for any number of time periods.

In order to check the logic of FAILMLE, the program was run treating the expected number of unit failures for a sample as the observed number of unit failures. The results (Table 7) indicate that the logic was correct. Differences between true values and the estimated values were due to round off errors.

Those calculations which were done by hand could be incorporated into the programs. The test of fit described in section 2.2 could be included in both MONTE and FAILMLE. Those calculations necessary to estimate the variance-covariance matrix of the parameter estimates found with MONTE could be included in the program. FAILMLE would be of more practical use if the program and its input were adjusted to predict point estimates and to determine confidence intervals for the reliability of new electronic units.

Figure 10

Input Format For The Program FAILMLE

```
10,3
27,3,450,  1000,368                    UNIT 1
25,10,350,  1500,488                   UNIT 2
31,0,375,  200,69                      UNIT 3
16,11,400,  1200,369                   UNIT 4
10,25,200,  500,106                    UNIT 5
35,0,425,  1800,698                    UNIT 6
29,29,325,  900,342                    UNIT 7
21,0,400,  2000,610                    UNIT 8
21,9,428,  1400,478                    UNIT 9
17,25,216,  1000,256                   UNIT 10
.0005
5
```

TABLE 7

RESULTS OF TEST RUN ON THE PROGRAM FAILMLE

Component Family	Estimated Component Family Failure Rate	True Component Family Failure Rate
1	10.0680	10.0
2	4.9885	5.0
3	.9970	1.0

Unit	Estimated Unit Failure Rate	True Unit Failure Rate
1	735.4426	735.0
2	650.5281	650.0
3	685.9751	685.0
4	614.7542	615.0
5	424.7893	425.0
6	776.0962	775.0
7	760.6575	760.0
8	610.2202	610.0
9	683.0325	683.0
10	511.2166	511.0

PROGRAM LISTING - FAILMLE

```
C
C THIS PROGRAM USES THE MAXIMUM LIKELIHOOD METHOD OF ESTIMATION AND
C THE METHOD OF SCORING TO DETERMINE THE FAILURE RATES OF COMPONENT
C FAMILIES.
C
      DIMENSION Q(50,25),XINUSE(50),FAIL(50),XINF(25,25),XLAMDA(50),
     $ LWORK(25),R(25,25),PREV(25),MWORK(25),ALPHA(25),FAC(25),
     $ S(25),CONDSD(25)
      READ(9,*)N,M
C
C N IS THE NUMBER OF ROWS (UNITS) & M IS THE NUMBER OF COLUMNS
C (COMPONENT FAMILIES).
C
      CALL SOLVE(Q,XINUSE,FAIL,LWORK,MWORK,ALPHA,M,N,FAC,S,XLAMDA,XINF,
     $ PREV,R,CONDSD)
      STOP
      END
C
      SUBROUTINE SOLVE(Q,XINUSE,FAIL,LWORK,MWORK,ALPHA,M,N,FAC,S,
     $ XLAMDA,XINF,PREV,R,CONDSD)
      DIMENSION Q(N,M),XINUSE(N),FAIL(N),XINF(M,M),S(M),LWORK(M),
     $ MWORK(M),R(M,M),PREV(M),ALPHA(M),FAC(M),XLAMDA(N),
     $ CONDSD(M)
C
      WRITE(8,200)
200   FORMAT(20X,'DATA MATRIX:',/,/)
C
      DO 10 I=1,N
      READ(9,*)(Q(I,J),J=1,M),XINUSE(I),FAIL(I)
      WRITE(8,210)(Q(I,J),J=1,M)
210   FORMAT(3X,25F4.0)
10    CONTINUE
C
      READ(9,*) TIME
      WRITE(8,220)
220   FORMAT(1H1)
      WRITE(8,230)
230   FORMAT(29X,'UNIT',15X,'NUMBER IN USE',10X,'NUMBER OF FAILURES',
     $ /)
C
      DO 15 I=1,N
      WRITE(8,240)I,XINUSE(I),FAIL(I)
240   FORMAT(30X,I2,2(20X,F5.0))
15    CONTINUE
C
      WRITE(8,250)TIME
250   FORMAT(/,/,40X,'TIME IN UNITS OF 10 MILLION HOURS =',F10.6)
C
C SET UP INITIAL VALUES.
C
      NIT=0
      READ(9,*) START
C
      DO 20 J=1,M
      ALPHA(J)=START
```

PROGRAM LISTING - FAILMLE (Cont'd)

```
20     CONTINUE
C
       WRITE(8,220)
       WRITE(8,260) NIT
260    FORMAT(/,15X,'N = ',I3)
       WRITE(8,270)(J,ALPHA(J),J=1,M)
270    FORMAT(25X,'ALPHA(',I2,') =',E16.8)
C
C BEGIN THE ITERATIVE PROCEDURE.
C
500    NIT=NIT+1
C
C WE WISH TO KEEP THE OLD VALUES OF ALPHA TO TEST FOR CONVERGENCE.
C
       DO 25 J=1,M
       PREV(J)=ALPHA(J)
25     CONTINUE
C
       WRITE(8,260)NIT
C
       CALL LAFAIL(N,M,XLAMDA,Q,ALPHA)
C
C DETERMINE THE SCORE VECTOR.
C
       DO 30 J=1,M
       S(J)=0.0
       DO 35 I=1,N
       F=FAIL(I)/XLAMDA(I) - TIME*XINUSE(I)
       S(J)=S(J)+Q(I,J)*F
35     CONTINUE
30     CONTINUE
C
       CALL INFAIL(N,M,Q,XINUSE,TIME,XLAMDA,XINF,LWORK,MWORK,
      $ CONDSD)
C
C DETERMINE THE ADJUSTMENT FACTOR
C
       DO 40 J=1,M
       FAC(J)=0.0
       DO 45 JJ=1,M
       FAC(J)=FAC(J)+XINF(J,JJ)*S(JJ)
45     CONTINUE
40     CONTINUE
C
C DETERMINE THE NEW ESTIMATES OF COMPONENT FAMILY FAILURE RATES.
C
       DO 50 J=1,M
       ALPHA(J)=ALPHA(J)+FAC(J)
50     CONTINUE
C
       WRITE(8,270)(J,ALPHA(J),J=1,M)
C
       DO 55 J=1,M
       IF(ABS(ALPHA(J)-PREV(J)).GT.0.000005) GOTO 520
55     CONTINUE
```

PROGRAM LISTING - FAILMLE (Cont'd)

```
C
      GOTO 510
C
520   IF(NIT.LT.100)GOTO 500
510   CALL LAFAIL(N,M,XLAMDA,Q,ALPHA)
      CALL INFAIL(N,M,Q,XINUSE,TIME,XLAMDA,XINF,LWORK,MWORK,
     $ CONDSD)
C
      WRITE(8,220)
      WRITE(8,280)
280   FORMAT(50X,'COVARIANCE MATRIX',/)
      WRITE(8,290)(J,J=1,M)
290   FORMAT(16(13X,I2),/)
C
      DO 60 I=1,M
      WRITE(8,300)I,(XINF(I,J),J=1,I)
300   FORMAT(2X,I2,16(2X,E13.7))
60    CONTINUE
C
C DETERMINE APPROXIMATE CONFIDENCE INTERVALS FOR OUR ESTIMATES OF
C THE FAILURE RATES.
C
      WRITE(8,220)
      WRITE(8,310)
310   FORMAT(25X,'APPROXIMATE 95% CONFIDENCE INTERVALS FOR COMPONENT',
     $ ' FAILURE RATES:',/,/)
      WRITE(8,320)
320   FORMAT(7X,'COMPONENT',6X,'ESTIMATED',8X,'UNCONDITIONAL',12X,
     $ 'CONDITIONAL')
      WRITE(8,330)
330   FORMAT(8X,'FAMILY',6X,'FAILURE RATE',7X,'VARIANCE C.I.',
     $ 12X,'VARIANCE C.I.',/)
      WRITE(8,340)
340   FORMAT(10X,2('-'),10X,8('-'),6X,19('-'),5X,19('-'))
C
      DO 65 J=1,M
      COMPLB=ALPHA(J)-1.96*SQRT(XINF(J,J))
      COMPUB=ALPHA(J)+1.96*SQRT(XINF(J,J))
      CONDLB=ALPHA(J)-1.96*CONDSD(J)
      CONDUB=ALPHA(J)+1.96*CONDSD(J)
      WRITE(8,350)J,ALPHA(J),COMPLB,COMPUB,CONDLB,CONDUB
350   FORMAT(10X,I2,10X,F8.5,':',5X,'(',F8.5,',',F8.5,')',
     $ 5X,'(',F8.5,',',F8.5,')')
65    CONTINUE
C
      WRITE(8,220)
C
C DETERMINE CONFIDENCE INTERVALS FOR THE UNIT FAILURE RATES FROM
C THE OBSERVED FAILURES AND FROM THE ESTIMATED FAILURE RATES.
C
      WRITE(8,360)
360   FORMAT(25X,'APPROXIMATE 95% CONFIDENCE INTERVALS FOR UNIT',
     $ ' FAILURE RATES',/,/)
      WRITE(8,370)
370   FORMAT(9X,'UNIT',17X,'FOR OBSERVED',40X,'FOR ESTIMATES',/,/)
```

PROGRAM LISTING - FAILMLE (Cont'd)

```
      WRITE(8,380)
380   FORMAT(10X,2('-'),5X,39('-'),15X,39('-'),/)
C
      DO 70 I=1,N
      PHI=FAIL(I)/(XINUSE(I)*TIME)
      POISLB=PHI-1.96*SQRT(PHI)
      POISUB=PHI+1.96*SQRT(PHI)
      ESTVAR=0.0
C
      DO 75 J=1,M
      DO 80 K=1,M
      ESTVAR=ESTVAR+Q(I,J)*Q(I,K)*XINF(J,K)
80    CONTINUE
75    CONTINUE
C
      ESTLB=XLAMDA(I)-1.96*SQRT(ESTVAR)
      ESTUB=XLAMDA(I)+1.96*SQRT(ESTVAR)
      WRITE(8,390)I,PHI,POISLB,POISUB,XLAMDA(I),ESTLB,ESTUB
390   FORMAT(10X,I2,5X ,F10.5,':',5X,'(',F10.5,',',F10.5,')',15X,F10.5,
     $ ':',5X,'(',F10.5,',',F10.5,')')
70    CONTINUE
C
C DETERMINE PRINCIPAL COMPONENTS, THAT IS BY CALCULATING THE
C EIGENVALUES AND THE EIGENVECTORS WE CAN DETERMINE NEW BASES
C FOR OUR COMPONENTS WHICH WILL NOT BE CORRELATED.
C
      CALL EIGEN(XINF,R,M,0)
      WRITE(8,220)
C
      DO 85 J=1,M
      WRITE (8,400)J,XINF(J,J)
400   FORMAT(/,20X,'EIGENVALUE(',I2,') = ',F16.8,/)
      WRITE(8,410)(J,I,R(I,J),I=1,M)
410   FORMAT(25X,'EIGENVECTOR(',I2,',',I2,') = ',E16.8)
85    CONTINUE
C
      RETURN
      END
```

PROGRAM LISTING - MONTE

```
C
C THIS PROGRAM USES THE MAXIMUM LIKELIHOOD METHOD OF ESTIMATION AND
C THE METHOD OF SCORING TO DETERMINE THE FAILURE RATES OF COMPONENT
C FAMILIES.
C
      DIMENSION Q(50,25),XINUSE(50),FAIL(50),XINF(25,25),XLAMDA(50),
     $ LWORK(25),R(25,25),PREV(25),MWORK(25),ALPHA(25),FAC(25),
     $ S(25),CONDSD(25)
      READ(9,*)N,M
C
C N IS THE NUMBER OF ROWS (UNITS) & M IS THE NUMBER OF COLUMNS
C (COMPONENT FAMILIES).
C
      CALL SOLVE(Q,XINUSE,FAIL,LWORK,MWORK,ALPHA,M,N,FAC,S,XLAMDA,XINF,
     $ PREV,R,CONDSD)
      STOP
      END
C
      SUBROUTINE SOLVE(Q,XINUSE,FAIL,LWORK,MWORK,ALPHA,M,N,FAC,S,
     $ XLAMDA,XINF,PREV,R,CONDSD)
      DIMENSION Q(N,M),XINUSE(N),FAIL(N),XINF(M,M),S(M),LWORK(M),
     $ CMPFL(3),RATE(10),XMEAN(10),S1(10),S2(10),
     $DEV(10),XBAR(3),SUM(3),
     $ MWORK(M),R(M,M),PREV(M),ALPHA(M),FAC(M),XLAMDA(N),
     $ CONDSD(M)
C
      WRITE(8,200)
200   FORMAT(20X,'DATA MATRIX:',/,/)
C
      DO 10 I=1,N
      READ(9,*)(Q(I,J),J=1,M),XINUSE(I),FAIL(I)
      WRITE(8,210)(Q(I,J),J=1,M)
210   FORMAT(3X,25F4.0)
10    CONTINUE
C
      READ(9,*) TIME
      WRITE(8,250)TIME
250   FORMAT(/,/,40X,'TIME IN UNITS OF 10 MILLION HOURS =',F10.6)
C
      WRITE(8,220)
220   FORMAT(1H1)
      CMPFL(1)=10.
      CMPFL(2)=5.
      CMPFL(3)=1.
C
      WRITE(8,221)
221   FORMAT(18X,'COMPONENT FAMILY',10X,'FAILURE RATE',/)
      WRITE(8,222)(J,CMPFL(J),J=1,M)
222   FORMAT(25X,I2,20X,F5.1)
C
C DETERMINE AND PRINT THE TRUE COMPONENT FAMILY FAILURE RATES.
C
      WRITE(8,220)
      WRITE(8,223)
223   FORMAT(20X,'UNIT',15X,'UNITS IN FIELD',13X,'TRUE FAILURE RATE',/)
```

PROGRAM LISTING - MONTE (Cont'd)

```
C
      DO 11 I=1,N
      RATE(I)=0.0
      DO 21 J=1,M
      RATE(I)=RATE(I)+Q(I,J)*CMPFL(J)
21    CONTINUE
      WRITE(8,224)I,XINUSE(I),FAIL(I)
224   FORMAT(25X,I2,20X,F5.0,20X,F8.0)
      XMEAN(I)=XINUSE(I)*TIME*RATE(I)
      DEV(I)=SQRT(XMEAN(I))
      S1(I)=0.0
      S2(I)=0.0
11    CONTINUE
C
      WRITE(8,220)
      IX=87
      DO 61 J=1,M
      XBAR(J)=0.0
      SUM(J)=0.0
61    CONTINUE
C
      DO 41 NSTEP = 1,100
C
C GENERATE APPROXIMATE POISSON RANDOM NUMBERS TO BE USED AS OBSERVED
C NUMBER OF UNIT FAILURES IN THE FIELD.
C
      DO 51 I=1,N
500   CALL GAUSS(IX,DEV(I),XMEAN(I),FAIL(I))
      IF(FAIL(I).LT.0.0)GOTO 500
51    CONTINUE
C
C SET UP INITIAL VALUES.
C
      NIT=0
C
      DO 20 J=1,M
      ALPHA(J)=CMPFL(J)
20    CONTINUE
C
C
C BEGIN THE ITERATIVE PROCEDURE.
C
600   NIT=NIT+1
C
C WE WISH TO KEEP THE OLD VALUES OF ALPHA TO TEST FOR CONVERGENCE.
C
      DO 25 J=1,M
      PREV(J)=ALPHA(J)
25    CONTINUE
C
C
      CALL LAFAIL(N,M,XLANDA,Q,ALPHA)
C
C DETERMINE THE SCORE VECTOR.
C
```

PROGRAM LISTING - MONTE (Cont'd)

```
      DO 30 J=1,M
      S(J)=0.0
      DO 35 I=1,N
      F=FAIL(I)/XLAMDA(I) - TIME*XINUSE(I)
      S(J)=S(J)+Q(I,J)*F
35    CONTINUE
30    CONTINUE
C
      CALL INFAIL(N,M,Q,XINUSE,TIME,XLAMDA,XINF,LWORK,MWORK,
     $ CONDSD)
C
C DETERMINE THE ADJUSTMENT FACTOR
C
      DO 40 J=1,M
      FAC(J)=0.0
      DO 45 JJ=1,M
      FAC(J)=FAC(J)+XINF(J,JJ)*S(JJ)
45    CONTINUE
40    CONTINUE
C
C DETERMINE THE NEW ESTIMATES OF COMPONENT FAMILY FAILURE RATES.
C
      DO 50 J=1,M
      ALPHA(J)=ALPHA(J)+FAC(J)
50    CONTINUE
C
C
      DO 55 J=1,M
      IF(ABS(ALPHA(J)-PREV(J)).GT.0.000005) GOTO 520
55    CONTINUE
C
      GOTO 510
C
520   IF(NIT.LT.100)GOTO 600
510   CALL LAFAIL(N,M,XLAMDA,Q,ALPHA)
      CALL INFAIL(N,M,Q,XINUSE,TIME,XLAMDA,XINF,LWORK,MWORK,
     $ CONDSD)
C
C DETERMINE APPROXIMATE CONFIDENCE INTERVALS FOR OUR ESTIMATES OF
C THE FAILURE RATES.
C
      WRITE(8,310)
310   FORMAT(25X,'APPROXIMATE 95% CONFIDENCE INTERVALS FOR COMPONENT',
     $ ' FAILURE RATES:',/,/)
      WRITE(8,320)
320   FORMAT(7X,'COMPONENT',6X,'ESTIMATED',8X,'UNCONDITIONAL',12X,
     $ 'CONDITIONAL')
      WRITE(8,330)
330   FORMAT(8X,'FAMILY',6X,'FAILURE RATE',7X,'VARIANCE C.I.',
     $ 12X,'VARIANCE C.I.',/)
      WRITE(8,340)
340   FORMAT(10X,2('-'),10X,8('-'),6X,19('-'),5X,19('-'))
C
      DO 65 J=1,M
      COMPLB=ALPHA(J)-1.96*SQRT(XINF(J,J))
```

```
      COMPUB=ALPHA(J)+1.96*SQRT(XINF(J,J))
      CONDLB=ALPHA(J)-1.96*CONDSD(J)
      CONDUB=ALPHA(J)+1.96*CONDSD(J)
      WRITE(8,350)J,ALPHA(J),COMPLB,COMPUB,CONDLB,CONDUB
350   FORMAT(10X,I2,10X,F8.5,':',5X,'(',F8.5,',',F8.5,')',
     $ 5X,'(',F8.5,',',F8.5,')')
      XBAR(J)=XBAR(J)+ALPHA(J)
      SUM(J)=SUM(J)+ALPHA(J)**2
65    CONTINUE
C
C
C DETERMINE CONFIDENCE INTERVALS FOR THE UNIT FAILURE RATES FROM
C THE OBSERVED FAILURES AND FROM THE ESTIMATED FAILURE RATES.
C
      WRITE(8,360)
360   FORMAT(25X,'APPROXIMATE 95% CONFIDENCE INTERVALS FOR UNIT',
     $ ' FAILURE RATES',/,/)
      WRITE(8,370)
370   FORMAT(9X,'UNIT',17X,'FOR OBSERVED',40X,'FOR ESTIMATES',/,/)
      WRITE(8,380)
380   FORMAT(10X,2('-'),5X,39('-'),15X,39('-'),/)
C
      DO 70 I=1,N
      PHI=FAIL(I)/(XINUSE(I)*TIME)
      POISLB=PHI-1.96*SQRT(PHI)
      POISUB=PHI+1.96*SQRT(PHI)
      ESTVAR=0.0
C
      DO 75 J=1,M
      DO 80 K=1,M
      ESTVAR=ESTVAR+Q(I,J)*Q(I,K)*XINF(J,K)
80    CONTINUE
75    CONTINUE
C
      ESTLB=XLAMDA(I)-1.96*SQRT(ESTVAR)
      ESTUB=XLAMDA(I)+1.96*SQRT(ESTVAR)
      WRITE(8,390)I,PHI,POISLB,POISUB,XLAMDA(I),ESTLB,ESTUB
390   FORMAT(10X,I2,5X ,F10.5,':',5X,'(',F10.5,',',F10.5,')',15X,F10.5,
     $ ':',5X,'(',F10.5,',',F10.5,')')
      S1(I)=S1(I)+XLAMDA(I)
      S2(I)=S2(I)+XLAMDA(I)**2
70    CONTINUE
C
41    CONTINUE
      WRITE(8,220)
      WRITE(8,375)
375   FORMAT(20X,'95% CONFIDENCE INTERVALS FOR THE ',
     $ 'COMPONENT FAILURE RATE ESTIMATES',/)
      WRITE(8,381)
381   FORMAT(17X,'COMPONENT',12X,'ESTIMATE',12X,
     $ 'CONFIDENCE INTERVAL',6X,'ACTUAL FAILURE RATE',/)
      WRITE(8,385)
385   FORMAT(20X,2('-'),15X,9('-'),11X,21('-'),10X,9('-'))
C
      DO 71 J=1,M
```

PROGRAM LISTING - MONTE (Cont'd)

```
      SUM(J)=(SUM(J)-((XBAR(J)**2)/100.))/99.
      XBAR(J)=XBAR(J)/100.
      XLB=XBAR(J)-1.96*SQRT(SUM(J))
      XUB=XBAR(J)+1.96*SQRT(SUM(J))
      WRITE(8,395)J,XBAR(J),XLB,XUB,CMPFL(J)
395   FORMAT(20X,I2,15X,F9.4,':',10X,'(',F9.4,',',F9.4,')',
     $ 10X,F9.5)
71    CONTINUE
C
      WRITE(8,220)
      WRITE(8,400)
400   FORMAT(20X,'95% CONFIDENCE INTERVALS FOR UNIT FAILURE RATE',
     $ ' ESTIMATES:',/)
      WRITE(8,410)
410   FORMAT(19X,'UNIT',15X,'ESTIMATE',12X,
     $ 'CONFIDENCE INTERVAL',/)
      WRITE(8,385)
C
      DO 81 I=1,N
      S2(I)=(S2(I)-((S1(I)**2)/100.))/99.
      S1(I)=S1(I)/100.
      XLB=S1(I)-1.96*SQRT(S2(I))
      XUB=S1(I)+1.96*SQRT(S2(I))
      WRITE(8,395)I,S1(I),XLB,XUB,RATE(I)
81    CONTINUE
C
      RETURN
      END
```

SUBROUTINE LISTING - INFAIL

```
      SUBROUTINE INFAIL(N,M,Q,XINUSE,TIME,XLAMDA,XINF,LWORK,MWORK,
     $ CONDSD)
C
C THIS SUBROUTINE DETERMINES THE INFORMATION MATRIX FOR THE FAILURE RATE
C PROBREM AND ITS INVERSE.
C
      DIMENSION Q(N,M),XINUSE(N),XINF(M,M),XLAMDA(N),LWORK(M),MWORK(M),
     $ A(325),CONDSD(M)
C
      DO 10 J=1,M
      DO 15 JJ=1,J
      XINF(J,JJ)=0.0
      DO 20 I=1,N
      XINF(J,JJ)=XINF(J,JJ)+TIME*XINUSE(I)*Q(I,J)*Q(I,JJ)/XLAMDA(I)
20    CONTINUE
15    CONTINUE
      CONDSD(J)=1.0/SQRT(XINF(J,J))
10    CONTINUE
C
      DO 25 J=2,M
      JMIN=J-1
      DO 30 JJ=1,JMIN
      XINF(JJ,J)=XINF(J,JJ)
30    CONTINUE
25    CONTINUE
C
      DO 35 J=1,M
35    CONTINUE
      LSUM=0
      DO 40 J=1,M
      DO 45 K=1,J
      LSUM=LSUM+1
      A(LSUM)=10.E7*XINF(J,K)
45    CONTINUE
40    CONTINUE
C
      EPS=10.E-6
      CALL SINV(A,M,EPS,IER)
C
      LSUM=0
      DO 50 J=1,M
      DO 55 K=1,J
      LSUM=LSUM+1
      XINF(J,K)=10.E7*A(LSUM)
      XINF(K,J)=XINF(J,K)
55    CONTINUE
50    CONTINUE
C
      DO 60 J=1,M
60    CONTINUE
      DO 65 J=1,M
65    CONTINUE
      RETURN
      END
```

SUBROUTINE LISTING - LAFAIL

```
      SUBROUTINE LAFAIL(N,M,XLAMDA,Q,ALPHA)
C
C  THIS SUBROUTINE DETERMINES ESTIMATES OF THE UNIT FAILURE RATES.
C
      DIMENSION XLAMDA(N),Q(N,M),ALPHA(M)
C
      DO 10 I=1,N
      XLAMDA(I)=0.0
      DO 15 J=1,M
      XLAMDA(I)=XLAMDA(I)+ALPHA(J)*Q(I,J)
15    CONTINUE
10    CONTINUE
C
      RETURN
      END
```

APPENDIX 4

Program for the Component Yield Problem

The computing work for the problem of estimating component family yields involved five programs. All are written in FORTRAN. The programs LINLSQ and CARLO use the method of weighted least squares. LINLSQ is designed for practical use while CARLO, which generates one hundred random samples of unit yields, is designed to study the properties of the estimates. The programs PREDMLE, COMB and SIMPLEX are designed to find maximum likelihood estimates of component family yields. PREDMLE uses a common starting value for the component family yield estimates and uses the method of scoring to find those estimates which maximize the likelihood. Using those estimates determined by the method of weighted least squares as starting values, the program COMB uses the method of scoring to find maximum likelihood estimates of component family yields. The program SIMPLEX uses the Nelder Mead Simplex method of finding maximum likelihood estimates of the component family yields. The method of weighted least squares is used to determine one of the vertices of the initial simplex used in SIMPLEX. The output file for all five programs has a length of one hundred and thirty-three columns including the carriage control.

Input for LINLSQ (Figure 11) is read from an external file having a length of eighty columns. It is unformatted and is general for any number of units observed and any number of component families. The first row of the input gives N, the number of distinct component families for which yields

can be estimated. The first M columns of the second to (N+1)st rows inclusive is the matrix $Q = [q_{ij}]_{N\times M}$ where q_{ij} is the number of components of the j^{th} family in a unit of type i. The (M+1)st entry in the (i+1)st row is the number of units of type i produced during the production period. The (M+2)nd entry in the (i+1)st row is the number of units of type i accepted during the production period. The (N+2)nd row gives the 97.5 percent critical value of the student's t distribution on N-M degrees of freedom.

LINLSQ calls three subroutines, GMPRD,SINV, and EIGEN, from the SSP FORTRAN library GMPRD determines the product of two general matrices. SINV calculates the inverse of a positive definite symmetric matrix and is used to determine the estimated variance-covariance matrix of the estimated component family yields. EIGEN is used to determine the eigenvectors and eigenvalues of that matrix.

The logic of LINLSQ was tested by using as input the expected number of acceptable units for a sample for which the component family yields are known. The program was adjusted to allow for the fact that there would be no variance in the input data. The results (Table 8) indicated that the logic was correct. Differences between the true values and the estimates were due to round off error.

Input for CARLO is the same as that for LINLSQ except that M is restricted to the value three. Input for the number of acceptable units of each type for one sample is required, but the results of the program would allow the omission of this input. CARLO could be

generalized to allow for any number of component families and for any number of generated samples. Three SSP subroutines GAUSS, GMPRD, and SINV are required for CARLO. GAUSS, a normal random number generator, is used to generate the number of acceptable units of each type.

Input for PREDMLE is the same as that for LINLSQ except that the entry in the (N+2)nd row gives a value which is given to the M component family yield estimates as starting values for the iterative procedure. PREDMLE calls the subroutine MINV from the SSP FORTRAN library in order to determine the inverse of the information matrix. The same data used to test LINLSQ was used to test the logic of PREDMLE. To start the iterative procedure in the test, the true component family yields were used as starting values. Test results (Table 9) indicate that the logic is correct.

The input for COMB is the same as that for LINLSQ. Five external subroutines are called from COMB. The subroutines SINV, GMPRD, and EIGEN are called from the SSP FORTRAN library. COMYLD calculates predicted unit yields from estimates of component family yields. COMINF calculates the estimated information matrix and its inverse.

The input for SIMPLEX is identical to that for LINLSQ except that no entry in the (N+2)nd row of the external data file is required. The subroutines GMPRD and SINV are called from the SSP FORTRAN library. As well, seven other subroutines, XCALC,XPLACE,XREF, XCONTR,XCENT,XSHR, and XPAND and the function XF are called externally

from SIMPLEX. The purposes of these subprograms are described in the initial comments of their listings.

With the exception of CARLO, all of the above programs were originally designed for practical use. Due to numerical problems, however, only LINLSQ can be used with any degree of success.

All approximate confidence intervals are calculated at the ninety-five percent confidence level. Alterations to the programs and their input could be made which would allow for different levels of confidence.

Those calculations which were done by hand using results from LINLSQ and CARLO could be incorporated into the programs. The test of fit described in section 3.3 could be included in both LINLSQ and CARLO. Those calculations necessary to estimate the variance-covariance matrix of the parameter estimates in CARLO could be included in the program. LINLSQ would be of more practical use if the program and its input were adjusted to predict point estimates and determine confidence intervals for the yields of newly designed electronic units.

Figure 11

Input Format For The Program LINLSQ

```
10,3
27,3,450,   1000,553          UNIT 1
25,10,350,  500,144           UNIT 2
31,0,375,   800,574           UNIT 3
16,11,400,   750,101          UNIT 4
10,25,200,   150,92           UNIT 5
35,0,425,   650,496           UNIT 6
29,29,325,   125,79           UNIT 7
21,0,400,   600,241           UNIT 8
21,9,428,   850,337           UNIT 9
17,25,216,   350,219          UNIT 10
2,365
```

TABLE 8

RESULTS OF TEST RUN ON THE PROGRAM LINLSQ

Component Family	Estimated Component Family Yield	True Component Family Yield
1	.9899	.9900
2	.9949	.9950
3	.9990	.9990

TABLE 9

RESULTS OF TEST RUN ON THE PROGRAM PREDMLE

Component Family	Estimated Component Family Yield	True Component Family Yield
1	.9900	.9900
2	.9950	.9950
3	.9990	.9990

PROGRAM LISTING - LINLSQ

```
C THIS PROGRAM TAKES THE LOGARITHM OF THE UNIT YIELDS TO FORM A LINEAR
C EQUATION IN THE LOGARITHMS OF THE COMPONENT YIELDS,  THESE EQUATIONS
C ARE SOLVED USING MULTIPLE REGRESSION,
C
      DIMENSION Q(50,25),Y(50),ADJYLD(50,1),QTRANS(25,50),CMPYLD(25),
     $ L(25),W(50),CONS(25,1),MWORK(25),NEW(50),DER(25,25),Z(25,50),
     $ A(325),R(25,25),QADJ(50,25),YIELD(50)
      REAL NEW
      READ(9,*) N,M
C
C N IS THE NUMBER OF ROWS (UNITS) & M IS THE NUMBER OF
C COLUMNS (COMPONENT FAMILIES),
C
      CALL OLS(N,MWORK,Q,YIELD,QTRANS,CMPYLD,L,M,NEW,
     $ DER,Z,CONS,Y,W,A,ADJYLD,QADJ,R)
      STOP
      END
C
      SUBROUTINE OLS(N,MWORK,Q,YIELD,QTRANS,CMPYLD,L,M,
     $ NEW,DER,Z,CONS,Y,W,A,ADJYLD,QADJ,R)
      DIMENSION Q(N,M),A(325),ADJYLD(N,1),QTRANS(M,N),
     $ CMPYLD(M),R(M,M),W(N),Y(N),L(M),MWORK(M),NEW(N),
     $ DER(M,M),QADJ(N,M),YIELD(N),Z(M,N),CONS(M,1)
      REAL NEW
C
      WRITE(8,200)
200   FORMAT(20X,'DATA MATRIX:',/,/)
C
      DO 10 I=1,N
      READ(9,*)(Q(I,J),J=1,M),NEW(I),Y(I)
      WRITE(8,210)(Q(I,J),J=1,M)
210   FORMAT(3X,25F4.0)
10    CONTINUE
C
      WRITE(8,220)
220   FORMAT(1H1)
      WRITE(8,230)
230   FORMAT(/,/)
C
C
C
C CALCULATE THE YIELDS AND ADJUST THE YIELDS AND THE Q MATRIX FOR
C UNEQUAL VARIANCES,  THEN PRINT OUT THE UNITS ACCEPTED, THE NUMBER
C TESTED, THE YIELD AND THE ADJUSTED YIELDS,
C
      WRITE(8,240)
240   FORMAT(14X,'UNIT',13X,'UNITS TESTED',15X,'UNITS ACCEPTED',
     $ 14X,'UNIT YIELD',13X,'ADJUSTED YIELD',/)
C
      DO 15 I=1,N
      YIELD(I)=Y(I)/NEW(I)
      W(I)=SQRT((NEW(I)-Y(I))/(NEW(I)*Y(I)))
      ADJYLD(I,1)=ALOG(YIELD(I))/W(I)
      DO 20 J=1,M
      QADJ(I,J)=Q(I,J)/W(I)
```

PROGRAM LISTING - LINLSQ (Cont'd)

```
20      CONTINUE
        WRITE(8,250)I,NEW(I),Y(I),YIELD(I),ADJYLD(I,1)
250     FORMAT(15X,I2,2(20X,F5.0),19X,F7.4,17X,F9.4)
15      CONTINUE
C
C DETERMINE THE TRANSPOSE OF THE ADJUSTED Q MATRIX.
C
        DO 25 I=1,N
        DO 30 J=1,M
        QTRANS(J,I)=QADJ(I,J)
30      CONTINUE
25      CONTINUE
C
C NOW WE CAN PERFORM THE ACTUAL REGRESSION.
C
        CALL GMPRD(QTRANS,QADJ,DER,M,N,M)
C
C THE MATRIX DER CONSISTS OF NON NEGATIVE ENTRIES.  HENCE WE CAN
C SCALE THE MATRIX SO THAT THE INVERSE OF A MATRIX WITH A SMALLER
C RANGE OF VALUES CAN BE DETERMINED.  THIS WILL REDUCE THE CHANCES
C OF THE PROGRAM CRASHING THROUGH A REGISTER OVERFLOW OR UNDERFLOW.
C
        ZZ=1000.*FLOAT(N)
        DO 35 J=1,M
        DO 40 JJ=1,M
        DER(J,JJ)=DER(J,JJ)/ZZ
40      CONTINUE
35      CONTINUE
C
C WE DETERMINE THE INVERSE OF DER USING THE SSP SUBROUTINE SINV
C WHICH DETERMINES THE INVERSE OF POSITIVE DEFINITE SYMMETRIC MATRICES.
C
        LSUM=0
        DO 45 J=1,M
        DO 50 K=1,J
        LSUM=LSUM+1
        A(LSUM)=DER(J,K)
50      CONTINUE
45      CONTINUE
C
        EPS=.001
        CALL SINV(A,M,EPS,IER)
C
        LSUM=0
        DO 55 J=1,M
        DO 60 K=1,J
        LSUM=LSUM+1
        DER(J,K)=A(LSUM)
        DER(K,J)=DER(J,K)
60      CONTINUE
55      CONTINUE
C
C NOW WE HAVE TO RESTORE DER TO ITS ORIGINAL SCALING BEFORE WE
C CAN USE IT.
C
```

PROGRAM LISTING - LINLSQ (Cont'd)

```
      DO 65 J=1,M
      DO 70 JJ=1,M
      DER(J,JJ)=DER(J,JJ)/ZZ
70    CONTINUE
65    CONTINUE
C
      CALL GMPRD(DER,QTRANS,Z,M,M,N)
      CALL GMPRD(Z,ADJYLD,CONS,M,N,1)
      WRITE(8,220)
      WRITE(8,230)
C
      SIGMA=0.0
      DO 75 I=1,N
      E=0.0
      DO 80 J=1,M
      E=E+QADJ(I,J)*CONS(J,1)
80    CONTINUE
      E=ADJYLD(I,1)-E
      SIGMA=SIGMA+E**2
75    CONTINUE
      NM=N-M
      SIGMA=SIGMA/FLOAT(NM)
      DO 85 J=1,M
      DO 90 JJ=1,M
      DER(J,JJ)=DER(J,JJ)*SIGMA
90    CONTINUE
85    CONTINUE
C
      WRITE(8,260)
260   FORMAT(50X,'COVARIANCE MATRIX',/)
      WRITE(8,270)(J,J=1,M)
270   FORMAT(16(13X,I2),/)
      DO 95 I=1,M
      WRITE(8,280)I,(DER(I,J),J=1,I)
280   FORMAT(2X,I2,16(2X,E13.7))
95    CONTINUE
      READ(9,*)T
C
C DETERMINE CONFIDENCE INTERVALS FOR THE ESTIMATES OF THE LOGARITHM
C OF THE COMPONENT YIELDS AND FOR THE COMPONENT YIELD ESTIMATES.
C
      WRITE(8,220)
      WRITE(8,230)
      WRITE(8,290)NM
290   FORMAT(25X,'APPROXIMATE 95% CONFIDENCE INTERVALS ON',I2,1X,
     $ 'DEGREES OF FREEDOM:',/)
      WRITE(8,300)
300   FORMAT(3X,'COMPONENT',12X,'LOG OF COMPONENT YIELD',43X,
     $ 'COMPONENT YIELD',/)
      WRITE(8,310)
310   FORMAT(10X,2('-'),5X,50('-'),15X,36('-'))
      DO 100 J=1,M
      XLB=CONS(J,1)-T*SQRT(DER(J,J))
      UB=CONS(J,1)+T*SQRT(DER(J,J))
      CMPYLD(J)=EXP(CONS(J,1))
```

PROGRAM LISTING - LINLSQ (Cont'd)

```
      EXPLB=EXP(XLB)
      EXPUB=EXP(UB)
      WRITE(8,320)J,CONS(J,1),XLB,UB,CMPYLD(J),EXPLB,EXPUB
320   FORMAT(10X,I2,5X,E14.6,5X,'(',E14.6,',',E14.6,')',15X,F8.5,5X,
     $ '(',E10.5,',',E10.5,')')
100   CONTINUE
C
C FIND APPROXIMATE CONFIDENCE INTERVALS FOR THE UNIT YIELDS USING
C BOTH THE OBSERVED YIELDS AND THE ESTIMATED YIELDS.
C
      WRITE(8,220)
      WRITE(8,230)
      WRITE(8,330)
330   FORMAT(25X,'APPROXIMATE 95% CONFIDENCE INTERVALS FOR UNIT ',
     $ 'YIELDS:',/,/)
      WRITE(8,340)
340   FORMAT(9X,'UNIT',14X,'FOR OBSERVED',35X,'FOR ESTIMATES',/)
      WRITE(8,350)
350   FORMAT(10X,2('='),5X,33('='),15X,37('='))
C
      DO 105 I=1,N
      FAC=YIELD(I)*(1.0-YIELD(I))/NEW(I)
      FAC=1.96*SQRT(FAC)
      BINLB=YIELD(I)-FAC
      BINUB=YIELD(I)+FAC
      ESTYLD=0.0
      ESTVAR=0.0
C
      DO 110 J=1,M
      ESTYLD = ESTYLD+Q(I,J)*CONS(J,1)
C
      DO 115 K=1,M
      ESTVAR=ESTVAR+Q(I,K)*Q(I,J)*DER(J,K)
115   CONTINUE
110   CONTINUE
C
      SD=SQRT(ESTVAR)
      ESTLB=EXP(ESTYLD-1.96*SD)
      ESTUB=EXP(ESTYLD+1.96*SD)
      ESTYLD=EXP(ESTYLD)
      WRITE(8,360)I,YIELD(I),BINLB,BINUB,ESTYLD,ESTLB,ESTUB
360   FORMAT(10X,I2,5X,F8.5,':',5X,'(',F8.5,',',F8.5,')',15X,
     $ F8.5,':',5X,'(',E10.5,',',E10.5,')')
105   CONTINUE
C
C DETERMINE PRINCIPAL COMPONENTS, THAT IS BY CALCULATING THE
C EIGENVECTORS, WE CAN DETERMINE NEW BASES FOR OUR COMPONENTS WHICH
C WILL BE INDEPENDENT.
C
      CALL EIGEN(DER,R,M,0)
      WRITE(8,220)
C
      DO 120 J=1,M
      WRITE(8,370)J,DER(J,J)
370   FORMAT(/,20X,'EIGENVALUE(',I2,') = ',E16.8,/)
```

PROGRAM LISTING - LINLSQ (Cont'd)

```
      WRITE(8,380)(J,I,R(I,J),I=1,M)
380   FORMAT(25X,'EIGENVECTOR(',I2,',',I2,') = ',E16.8)
120   CONTINUE
C
      RETURN
      END
```

PROGRAM LISTING - CARLO

```
C THIS PROGRAM GENERATES 100 SAMPLES OF YIELD DATA AND COMPUTES
C CONFIDENCE INTERVALS FOR THE ESTIMATES  USING THE RESULTS OF ALL
C OF THE SAMPLES.  THE YIELD ESTIMATES FOR EACH SAMPLE IS FOUND USING
C THE METHOD OF THE PROGRAM LINLSQ.
C
      DIMENSION Q(50,25),Y(50),ADJYLD(50,1),QTRANS(25,50),CMPYLD(25),
     $ P(50),AC(25),XMEAN(50),DEV(50),XBAR(25),
     $ SUM(25),S1(50),S2(50),
     $ L(25),W(50),CONS(25,1),MWORK(25),NEW(50),DER(25,25),Z(25,50),
     $ A(325),R(25,25),QADJ(50,25),YIELD(50)
      REAL NEW
      READ(9,*) N,M
C
C N IS THE NUMBER OF ROWS (UNITS) & M IS THE NUMBER OF
C COLUMNS (COMPONENT FAMILIES).
C
      CALL OLS(N,MWORK,Q,YIELD,QTRANS,CMPYLD,L,M,NEW,
     $ P,AC,XMEAN,DEV,XBAR,SUM,S1,S2,
     $ DER,Z,CONS,Y,W,A,ADJYLD,QADJ,R)
      STOP
      END
C
      SUBROUTINE OLS(N,MWORK,Q,YIELD,QTRANS,CMPYLD,L,M,
     $ NEW,P,AC,XMEAN,DEV,XBAR,SUM,S1,S2,
     $ DER,Z,CONS,Y,W,A,ADJYLD,QADJ,R)
      DIMENSION Q(N,M),A(325),ADJYLD(N,1),QTRANS(M,N),
     $ CMPYLD(M),R(M,M),W(N),Y(N),L(M),MWORK(M),NEW(N),
     $ P(N),AC(M),XMEAN(N),DEV(N),XBAR(M),SUM(M),
     $ S1(10),S2(10),
     $ DER(M,M),QADJ(N,M),YIELD(N),Z(M,N),CONS(M,1)
      REAL NEW
C
      WRITE(6,200)
200   FORMAT(20X,'DATA MATRIX:',/,/)
C
      DO 10 I=1,N
      READ(9,*)(Q(I,J),J=1,M),NEW(I),Y(I)
      WRITE(6,210)(Q(I,J),J=1,M)
210   FORMAT(3X,25F4.0)
10    CONTINUE
C
      WRITE(6,220)
220   FORMAT(1H1)
C
      AC(1)=.99
      AC(2)=.995
      AC(3)=.999
C
      WRITE(6,231)
231   FORMAT(22X,'COMPONENT',16X,'YIELD',/)
      WRITE(6,232)(J,AC(J),J=1,M)
232   FORMAT(25X,I2,20X,F5.3)
C
C DETERMINE AND PRINT THE TRUE UNIT YIELDS.
C
```

PROGRAM LISTING - CARLO (Cont'd)

```
      WRITE(6,220)
      WRITE(6,240)
240   FORMAT(24X,'UNIT',16X,'UNITS TESTED',17X,
     $ 'TRUE UNIT YIELD',/)
C
      DO 11 I=1,N
      P(I)=0.0
      DO 21 J=1,M
      P(I)=P(I)+Q(I,J)*ALOG(AC(J))
21    CONTINUE
      P(I)=EXP(P(I))
      WRITE(6,250)I,NEW(I),P(I)
250   FORMAT(25X,I2,20X,F5.0,20X,F10.8)
      XMEAN(I)=P(I)*NEW(I)
      DEV(I)=SQRT(XMEAN(I)*(1.0-P(I)))
      S1(I)=0.0
      S2(I)=0.0
11    CONTINUE
C
      WRITE(6,220)
      IX=79
      READ(9,*)T
C
      DO 61 J=1,M
      XBAR(J)=0.0
      SUM(J)=0.0
61    CONTINUE
C
      DO 41 NSTEP=1,100
C
C GENERATE APPROXIMATE BINOMIAL OBSERVATIONS TO BE USED AS
C THE NUMBER OF UNITS ACCEPTED.
C
      DO 51 I=1,N
500   CALL GAUSS(IX,DEV(I),XMEAN(I),Y(I))
      IF(Y(I).LT.0.0)GOTO 500
      IF(Y(I).GT.NEW(I))GOTO 500
51    CONTINUE
C
C CALCULATE THE YIELDS AND ADJUST THE YIELDS AND THE Q MATRIX FOR
C UNEQUAL VARIANCES.  THEN PRINT OUT THE UNITS ACCEPTED, THE NUMBER
C TESTED, THE YIELD AND THE ADJUSTED YIELDS.
C
      DO 15 I=1,N
      YIELD(I)=Y(I)/NEW(I)
      W(I)=SQRT((NEW(I)-Y(I))/(NEW(I)*Y(I)))
      ADJYLD(I,1)=ALOG(YIELD(I))/W(I)
      DO 20 J=1,M
      QADJ(I,J)=Q(I,J)/W(I)
20    CONTINUE
15    CONTINUE
C
C DETERMINE THE TRANSPOSE OF THE ADJUSTED Q MATRIX.
C
      DO 25 I=1,N
```

PROGRAM LISTING - CARLO (Cont'd)

```
      DO 30 J=1,M
      QTRANS(J,I)=QADJ(I,J)
30    CONTINUE
25    CONTINUE
C
C NOW WE CAN PERFORM THE ACTUAL REGRESSION.
C
      CALL GMPRD(QTRANS,QADJ,DER,M,N,M)
C
C THE MATRIX DER CONSISTS OF NON NEGATIVE ENTRIES.  HENCE WE CAN
C SCALE THE MATRIX SO THAT THE INVERSE OF A MATRIX WITH A SMALLER
C RANGE OF VALUES CAN BE DETERMINED.  THIS WILL REDUCE THE CHANCES
C OF THE PROGRAM CRASHING THROUGH A REGISTER OVERFLOW OR UNDERFLOW.
C
      ZZ=1000.*FLOAT(N)
      DO 35 J=1,M
      DO 40 JJ=1,M
      DER(J,JJ)=DER(J,JJ)/ZZ
40    CONTINUE
35    CONTINUE
C
C WE DETERMINE THE INVERSE OF DER USING THE SSP SUBROUTINE SINV
C WHICH DETERMINES THE INVERSE OF POSITIVE DEFINITE SYMMETRIC MATRICES.
C
      LSUM=0
      DO 45 J=1,M
      DO 50 K=1,J
      LSUM=LSUM+1
      A(LSUM)=DER(J,K)
50    CONTINUE
45    CONTINUE
C
      EPS=.001
      CALL SINV(A,M,EPS,IER)
C
      LSUM=0
      DO 55 J=1,M
      DO 60 K=1,J
      LSUM=LSUM+1
      DER(J,K)=A(LSUM)
      DER(K,J)=DER(J,K)
60    CONTINUE
55    CONTINUE
C
C NOW WE HAVE TO RESTORE DER TO ITS ORIGINAL SCALING BEFORE WE
C CAN USE IT.
C
      DO 65 J=1,M
      DO 70 JJ=1,M
      DER(J,JJ)=DER(J,JJ)/ZZ
70    CONTINUE
65    CONTINUE
C
      CALL GMPRD(DER,QTRANS,Z,M,M,N)
      CALL GMPRD(Z,ADJYLD,CONS,M,N,1)
```

PROGRAM LISTING - CARLO (Cont'd)

```
C
      SIGMA=0.0
      DO 75 I=1,N
      E=0.0
      DO 80 J=1,M
      E=E+QADJ(I,J)*CONS(J,1)
80    CONTINUE
      E=ADJYLD(I,1)-E
      SIGMA=SIGMA+E**2
75    CONTINUE
      NM=N-M
      SIGMA=SIGMA/FLOAT(NM)
      DO 85 J=1,M
      DO 90 JJ=1,M
      DER(J,JJ)=DER(J,JJ)*SIGMA
90    CONTINUE
85    CONTINUE
C
C
C DETERMINE CONFIDENCE INTERVALS FOR THE ESTIMATES OF THE LOGARITHM
C OF THE COMPONENT YIELDS AND FOR THE COMPONENT YIELD ESTIMATES.
C
      WRITE(6,290)NM
290   FORMAT(25X,'APPROXIMATE 95% CONFIDENCE INTERVALS ON',I2,1X,
     $ 'DEGREES OF FREEDOM:',/)
      WRITE(6,300)
300   FORMAT(3X,'COMPONENT',12X,'LOG OF COMPONENT YIELD',43X,
     $ 'COMPONENT YIELD',/)
      WRITE(6,310)
310   FORMAT(10X,2('='),5X,50('='),15X,36('='))
      DO 100 J=1,M
      XLB=CONS(J,1)-T*SQRT(DER(J,J))
      UB=CONS(J,1)+T*SQRT(DER(J,J))
      CMPYLD(J)=EXP(CONS(J,1))
      XBAR(J)=XBAR(J)+CMPYLD(J)
      SUM(J)=SUM(J)+CMPYLD(J)**2
      EXPLB=EXP(XLB)
      EXPUB=EXP(UB)
      WRITE(6,320)J,CONS(J,1),XLB,UB,CMPYLD(J),EXPLB,EXPUB
320   FORMAT(10X,I2,5X,E14.6,5X,'(',E14.6,',',E14.6,')',15X,F8.5,5X,
     $ '(',E10.5,',',E10.5,')')
100   CONTINUE
C
C FIND APPROXIMATE CONFIDENCE INTERVALS FOR THE UNIT YIELDS USING
C BOTH THE OBSERVED YIELDS AND THE ESTIMATED YIELDS.
C
      WRITE(6,220)
      WRITE(6,330)
330   FORMAT(25X,'APPROXIMATE 95% CONFIDENCE INTERVALS FOR UNIT ',
     $ 'YIELDS:',/,/)
      WRITE(6,340)
340   FORMAT(9X,'UNIT',14X,'FOR OBSERVED',35X,'FOR ESTIMATES',/)
      WRITE(6,350)
350   FORMAT(10X,2('='),5X,33('='),15X,37('='))
C
```

PROGRAM LISTING - CARLO (Cont'd)

```
      DO 105 I=1,N
      FAC=YIELD(I)*(1.0-YIELD(I))/NEW(I)
      FAC=1.96*SQRT(FAC)
      BINLB=YIELD(I)-FAC
      BINUB=YIELD(I)+FAC
      ESTYLD=0.0
      ESTVAR=0.0
C
      DO 110 J=1,M
      ESTYLD = ESTYLD+Q(I,J)*CONS(J,1)
C
      DO 115 K=1,M
      ESTVAR=ESTVAR+Q(I,K)*Q(I,J)*DER(J,K)
115   CONTINUE
110   CONTINUE
C
      SD=SQRT(ESTVAR)
      ESTLB=EXP(ESTYLD-1.96*SD)
      ESTUB=EXP(ESTYLD+1.96*SD)
      ESTYLD=EXP(ESTYLD)
      WRITE(6,360)I,YIELD(I),BINLB,BINUB,ESTYLD,ESTLB,ESTUB
360   FORMAT(10X,I2,5X,F8.5,':',5X,'(',F8.5,',',F8.5,')',15X,
     $ F8.5,':',5X,'(',E10.5,',',E10.5,')')
      S1(I)=S1(I)+ESTYLD
      S2(I)=S2(I)+ESTYLD**2
105   CONTINUE
C
41    CONTINUE
      WRITE(6,220)
      WRITE(6,370)
370   FORMAT(20X,'95% APPROXIMATE CONFIDENCE INTERVALS FOR THE ',
     $ 'COMPONENT YIELD ESTIMATES',/)
      WRITE(6,380)
380   FORMAT(17X,'COMPONENT',12X,'ESTIMATE',12X,
     $ 'CONFIDENCE INTERVAL',/)
      WRITE(6,385)
385   FORMAT(20X,2('-'),15X,9('-'),11X,21('-'))
C
      DO 71 J=1,M
      SUM(J)=(SUM(J)-((XBAR(J)**2)/100.))/99.
      XBAR(J)=XBAR(J)/100.
      XLB=XBAR(J)-1.96*SQRT(SUM(J))
      XUB=XBAR(J)+1.96*SQRT(SUM(J))
      WRITE(6,390)J,XBAR(J),XLB,XUB
390   FORMAT(20X,I2,15X,F9.6,':',10X,'(',F9.6,',',F9.6,')')
71    CONTINUE
C
      WRITE(6,220)
      WRITE(6,400)
400   FORMAT(20X,'95% CONFIDENCE INTERVALS FOR UNIT YIELD ',
     $ 'ESTIMATES',/)
      WRITE(6,410)
410   FORMAT(19X,'UNIT',15X,'ESTIMATE',12X,
     $ 'CONFIDENCE INTERVAL',/)
      WRITE(6,385)
```

PROGRAM LISTING - CARLO (Cont'd)

```
      DO 81 I=1,N
      S2(I)=(S2(I)-((S1(I)**2)/100.))/99.
      S1(I)=S1(I)/100.
      XLB=S1(I)-1.96*SQRT(S2(I))
      XUB=S1(I)+1.96*SQRT(S2(I))
      WRITE(6,390)I,S1(I),XLB,XUB
81    CONTINUE
C
      RETURN
      END
```

PROGRAM LISTING - PREDMLE

```
C
C THIS PROGRAM USES THE MAXIMUM LIKELIHOOD ESTIMATION METHOD AND THE
C METHOD OF "SCORING" TO DETERMINE THE YIELD OF COMPONENTS DURING
C PRODUCTION.
C
      DIMENSION Q(35,25),NEW(35),YIELD(35),Y(35),AHAT(25),
     U INF(25,25),L(25),MM(25),S(35),FAC(25),K(35),
     U BB(35,25,25),PREV(25)
      REAL NEW,INF,K
      READ(9,*)N,M
C
C N IS THE NUMBER OF ROWS (UNITS) & M IS THE NUMBER OF COLUMNS
C (COMPONENT FAMILIES).
C
      CALL SCORE(N,M,Q,NEW,YIELD,Y,AHAT,PREV,INF,L,MM,S,FAC,K,BB)
      STOP
      END
C
      SUBROUTINE SCORE(N,M,Q,NEW,YIELD,Y,AHAT,PREV,INF,L,MM,S,FAC,K,
     U BB)
      DIMENSION Q(N,M),NEW(N),YIELD(N),Y(N),AHAT(M),PREV(M),INF(M,M),
     U L(M),MM(M),S(N),FAC(M),K(N),BB(N,M,M)
      REAL NEW,INF,K
C
      DO 35 I=1,N
      READ(9,*)(Q(I,J),J=1,M),NEW(I),Y(I)
35    CONTINUE
C
C
C DETERMINE CONSTANTS WHICH WILL BE USED THROUGHOUT THE ITERATIVE
C PROCEDURE.
C
      DO 21 I=1,N
      DO 31 KK=1,M
      DO 41 LL=1,KK
      BB(I,KK,LL)=NEW(I)*Q(I,KK)*Q(I,LL)
41    CONTINUE
31    CONTINUE
21    CONTINUE
C
C INITIALIZE THE VALUES
C
      MMM=M-1
C
      NIT=0
      READ(9,*) START
      DO 85 J=1,M
      AHAT(J)=START
85    CONTINUE
C
      WRITE(7,500)NIT
500   FORMAT(/,7X,'NIT = ',I3)
      WRITE(7,550)(J,AHAT(J),J=1,M)
550   FORMAT(15X,'AHAT(',I2,') = ',E13.6)
C
```

PROGRAM LISTING - PREDMLE (Cont'd)

```
C NOW WE BEGIN THE ITERATIVE PROCEDURE.
C
10     NIT=NIT+1
C
       DO 95 J=1,M
       PREV(J)=AHAT(J)
95     CONTINUE
C
C DETERMINE THE ESTIMATES OF THE UNIT ACCEPTANCE PROBABILITES.
C
       DO 205 I=1,N
       K(I)=0.0
       DO 215 J=1,M
       K(I)=K(I)+G(I,J)*ALOG(AHAT(J))
215    CONTINUE
       K(I)=EXP(K(I))
205    CONTINUE
C
C DETERMINE THE INFORMATION MATRIX AND ITS INVERSE.
C
       DO 105 KK=1,M
       DO 115 LL=1,KK
       INF(KK,LL)=0.0
       DO 100 I=1,N
       INF(KK,LL)=INF(KK,LL)+BB(I,KK,LL)*K(I)/(1.0-K(I))
100    CONTINUE
       INF(KK,LL)=INF(KK,LL)/(AHAT(KK)*AHAT(LL))
115    CONTINUE
105    CONTINUE
C
       DO 110 KK=1,MMM
       III=KK+1
       DO 120 LL=III,M
       INF(KK,LL)=INF(LL,KK)
120    CONTINUE
110    CONTINUE
C
       CALL MINV(INF,M,D,L,MM)
C
C DETERMINE THE SCORE VECTOR.
C
       DO 125 J=1,M
       B=0.0
       DO 135 I=1,N
       B=Q(I,J)*(Y(I)-NEW(I)*K(I))
       B=B/(AHAT(J)*(1.0-K(I)))
       S(J)=S(J)+B
135    CONTINUE
125    CONTINUE
C
C DETERMINE THE ADJUSTMENT FACTOR
C
       DO 165 J=1,M
       FAC(J)=0.0
       DO 175 JJ=1,M
```

PROGRAM LISTING - PREDMLE (Cont'd)

```
      FAC(J)=FAC(J)+INF(J,JJ)*S(JJ)
175   CONTINUE
165   CONTINUE
C
C DETERMINE THE NEW VALUE OF AHAT.
C
      DO 185 J=1,M
      AHAT(J)=AHAT(J)+FAC(J)
185   CONTINUE
C
      DO 225 J=1,M
      IF(AHAT(J).LT.0.75)AHAT(J)=0.75
      IF(AHAT(J).GT.1.25)AHAT(J)=1.25
225   CONTINUE
      WRITE(7,500) NIT
      WRITE(7,550)(J,AHAT(J),J=1,M)
C
      DO 195 J=1,M
      IF(ABS(AHAT(J)-PREV(J)).GT.0.0000005) GOTO 20
195   CONTINUE
      RETURN
C
20    IF(NIT.LT.75)GOTO 10
      RETURN
      END
```

PROGRAM LISTING - COMB

```
C THIS PROGRAM USES THE METHOD OF THE PROGRAM LINLSQ TO FIND ESTIMATES
C OF THE COMPONENT YIELDS WHICH CAN THEN BE USED AS STARTING VALUES FOR
C FINDING MAXIMUM LIKELIHOOD ESTIMATES USING THE METHOD OF SCORING.
C
      DIMENSION Q(40,20),Y(40),ADJYLD(40,1),QTRANS(20,40),CMPYLD(20),
     $ BB(40,20,20),CONDSD(20),S(20),FAC(20),
     $ LWORK(20),W(40),CONS(20,1),MWORK(20),NEW(40),DER(20,20),Z(20,40),
     $ A(320),P(20,20),QADJ(40,20),YIELD(40)
      REAL NEW
      READ(9,*) N,M
C
C N IS THE NUMBER OF ROWS (UNITS) & M IS THE NUMBER OF
C COLUMNS (COMPONENT FAMILIES).
C
      CALL OLS(N,M,Q,YIELD,QTRANS,CMPYLD,LWORK,MWORK,NEW,
     $ DER,Z,CONS,Y,W,A,ADJYLD,QADJ,R,BB,CONDSD,S,FAC)
      STOP
      END
C
      SUBROUTINE OLS(N,M,Q,YIELD,QTRANS,CMPYLD,LWORK,MWORK,
     $ NEW,DER,Z,CONS,Y,W,A,ADJYLD,QADJ,R,BB,CONDSD,S,FAC)
      DIMENSION Q(N,M),A(325),ADJYLD(N,1),QTRANS(M,N),
     $ BB(N,M,M),CONDSD(M),S(M),FAC(M),
     $ CMPYLD(M),R(M,M),W(N),Y(N),LWORK(M),MWORK(M),NEW(N),
     $ DER(M,M),QADJ(N,M),YIELD(N),Z(M,N),CONS(M,1)
      REAL NEW
C
      WRITE(6,200)
200   FORMAT(20X,'DATA MATRIX:',/,/)
C
      DO 10 I=1,N
      READ(9,*) (Q(I,J),J=1,M),NEW(I),Y(I)
      WRITE(6,210) (Q(I,J),J=1,M)
210   FORMAT(3X,25F4.0)
10    CONTINUE
C
      WRITE(6,220)
220   FORMAT(1H1)
      WRITE(6,230)
230   FORMAT(/,/)
C
C CALCULATE THE YIELDS AND ADJUST THE YIELDS AND THE Q MATRIX FOR
C UNEQUAL VARIANCES.  THEN PRINT OUT THE UNITS ACCEPTED, THE NUMBER
C TESTED AND THE YIELDS.
C
      WRITE(6,240)
240   FORMAT(14X,'UNIT',13X,'UNITS TESTED',15X,'UNITS ACCEPTED',
     $ 14X,'UNIT YIELD',/)
C
      DO 15 I=1,N
      YIELD(I)=Y(I)/NEW(I)
      W(I)=SQRT((NEW(I)-Y(I))/(NEW(I)*Y(I)))
      ADJYLD(I,1)=ALOG(YIELD(I))/W(I)
      DO 20 J=1,M
      QADJ(I,J)=Q(I,J)/W(I)
```

PROGRAM LISTING - COMB (Cont'd)

```
20     CONTINUE
       WRITE(6,250)I,NEW(I),Y(I),YIELD(I)
250    FORMAT(15X,I2,2(20X,F5.0),19X,F7.4)
15     CONTINUE
C
C DETERMINE THE TRANSPOSE OF THE ADJUSTED Q MATRIX.
C
       DO 25 I=1,N
       DO 30 J=1,M
       QTRANS(J,I)=QADJ(I,J)
30     CONTINUE
25     CONTINUE
C
C NOW WE CAN PERFORM THE ACTUAL REGRESSION.
C
       CALL GMPRD(QTRANS,QADJ,DER,M,N,M)
C
C THE MATRIX DER CONSISTS OF NON NEGATIVE ENTRIES.  HENCE WE CAN
C SCALE THE MATRIX SO THAT THE INVERSE OF A MATRIX WITH A SMALLER
C RANGE OF VALUES CAN BE DETERMINED.  THIS WILL REDUCE THE CHANCES
C OF THE PROGRAM CRASHING THROUGH A REGISTER OVERFLOW OR UNDERFLOW.
C
       ZZ=1000.*FLOAT(N)
       DO 35 J=1,M
       DO 40 JJ=1,M
       DER(J,JJ)=DER(J,JJ)/ZZ
40     CONTINUE
35     CONTINUE
C
C WE DETERMINE THE INVERSE OF DER USING THE SSP SUBROUTINE SINV
C WHICH DETERMINES THE INVERSE OF POSITIVE DEFINITE SYMMETRIC MATRICES.
C
       LSUM=0
       DO 45 J=1,M
       DO 50 K=1,J
       LSUM=LSUM+1
       A(LSUM)=DER(J,K)
50     CONTINUE
45     CONTINUE
C
       EPS=.001
       CALL SINV(A,M,EPS,IER)
C
       LSUM=0
       DO 55 J=1,M
       DO 60 K=1,J
       LSUM=LSUM+1
       DER(J,K)=A(LSUM)
       DER(K,J)=DER(J,K)
60     CONTINUE
55     CONTINUE
C
C NOW WE HAVE TO RESTORE DER TO ITS ORIGINAL SCALING BEFORE WE
C CAN USE IT.
C
```

PROGRAM LISTING - COMB (Cont'd)

```
      DO 65 J=1,M
      DO 70 JJ=1,M
      DER(J,JJ)=DER(J,JJ)/ZZ
70    CONTINUE
65    CONTINUE
C
      CALL GMPRD(DER,QTRANS,Z,M,M,N)
      CALL GMPRD(Z,ADJYLD,CONS,M,N,1)
      WRITE(6,220)
      WRITE(6,230)
C
      DO 19 J=1,M
      CMPYLD(J)=EXP(CONS(J,1))
19    CONTINUE
C
C
C DETERMINE THE CONSTANTS WHICH WILL BE USED THROUGHOUT THE ITERATIVE
C PROCEDURE.
C
      DO 21 I=1,N
      DO 31 KK=1,M
      DO 41 LL=1,KK
      BB(I,KK,LL)=NEW(I)*Q(I,KK)*Q(I,LL)
41    CONTINUE
31    CONTINUE
21    CONTINUE
C
C INITIALIZE THE VALUES
C
      MMM=M-1
      NIT=0
      WRITE(6,600)NIT
600   FORMAT(/,7X,'NIT = ',I3)
      WRITE(6,550)(J,CMPYLD(J),J=1,M)
550   FORMAT(15X,'CMPYLD(',I2,') = ',E13.6)
C
C NOW BEGIN THE ITERATIVE PROCEDURE
C
500   NIT=NIT+1
      DO 95 J=1,M
      CONS(J,1)=CMPYLD(J)
95    CONTINUE
C
      CALL COMYLD(W,Q,M,N,CMPYLD)
      CALL COMINF(BB,M,N,DER,W,CONDSD,LWORK,MWORK,CMPYLD)
C
C DETERMINE THE SCORE VECTOR.
C
      DO 125 J=1,M
      B=0.0
      DO 135 I=1,N
      B=Q(I,J)*(Y(I)-NEW(I)*W(I))
      B=B/(CMPYLD(J)*(1.0-W(I)))
      S(J)=S(J)+B
135   CONTINUE
```

PROGRAM LISTING - COMB (Cont'd)

```
125   CONTINUE
C
      DO 235 J=1,M
      FAC(J)=0.0
      DO 245 JJ=1,M
      FAC(J)=FAC(J)+DER(J,JJ)*S(JJ)
245   CONTINUE
235   CONTINUE
C
C DETERMINE THE NEW ESTIMATE OF THE COMPONENT YIELD
C
      DO 185 J=1,M
      CMPYLD(J)=CMPYLD(J)+FAC(J)
185   CONTINUE
C
      WRITE(6,600)NIT
      WRITE(6,550) (J,CMPYLD(J),J=1,M)
C
      DO 195 J=1,M
      IF(ABS(CMPYLD(J)-CONS(J,1)).GT.0.00005) GOTO 510
195   CONTINUE
      RETURN
C
510   IF(NIT.LT.150)GOTO 500
      IF(NIT.EQ.250)RETURN
      WRITE(6,260)
260   FORMAT(50X,'COVARIANCE MATRIX',/)
      WRITE(6,270) (J,J=1,M)
270   FORMAT(16(13X,I2),/)
      DO 96 I=1,M
      WRITE(6,280)I,(DER(I,J),J=1,I)
280   FORMAT(2X,I2,16(2X,E13.7))
96    CONTINUE
      READ(9,*)T
C
C DETERMINE APPROXIMATE CONFIDENCE INTERVALS FOR OUR COMPONENT
C FAMILY YIELD ESTIMATES.
C
      WRITE(6,220)
      WRITE(6,230)
      WRITE(6,290)
290   FORMAT(25X,'APPROXIMATE 95% CONFIDENCE INTERVALS FOR',
     $ ' COMPONENT YIELDS',/,/)
      WRITE(6,300)
300   FORMAT(7X,'COMPONENT',6X,'ESTIMATED',8X,'UNCONDITIONAL',
     $ 16X,'CONDITIONAL')
      WRITE(6,305)
305   FORMAT(8X,'FAMILY',10X,'YIELD',10X,'VARIANCE C.I.',
     $ 16X,'VARIANCE C.I.',/)
      WRITE(6,310)
310   FORMAT(10X,2('-'),10X,8('-'),6X,19('-'),5X,23('-'))
C
      DO 100 J=1,M
      COMPLB=CMPYLD(J)-1.96*SQRT(DER(J,J))
      COMPUB=CMPYLD(J)+SQRT(DER(J,J))
```

PROGRAM LISTING - COMB (Cont'd)

```
      CONDLB=CMPYLD(J)-1.96*CONDSD(J)
      CONDUB=CMPYLD(J)+1.96*CONDSD(J)
      WRITE(6,390)J,CMPYLD(J),COMPLB,COMPUB,CONDLB,CONDUB
390   FORMAT(10X,I2,10X,F8.5,':',5X,'(',F8.5,',',F8.5,')',
     $ 5X,'(',F10.5,',',F10.5,')')
100   CONTINUE
C
C PRINT OUT THE ESTIMATED UNIT YIELDS AND COMPARE WITH THAT OBSERVED.
C
      WRITE(6,220)
      WRITE(6,230)
      WRITE(6,330)
330   FORMAT(9X,'UNIT',11X,'YIELD ESTIMATE',9X,'OBSERVED YIELD',
     $ 4X,'95% C.I. FOR OBSERVED YIELD',/)
      WRITE(6,410)
410   FORMAT(10X,2('-'),2(15X,8('-')),15X,19('-'))
C
      DO 105 I=1,N
      FACT=YIELD(I)*(1.0-YIELD(I))/NEW(I)
      FACT=1.96*SQRT(FACT)
      BINLB=YIELD(I)-FACT
      BINUB=YIELD(I)+FACT
      WRITE(6,420)I,W(I),YIELD(I),BINLB,BINUB
420   FORMAT(10X,I2,2(15X,F8.5),15X,'(',F8.5,',',F8.5,')')
105   CONTINUE
C
C DETERMINE PRINCIPAL COMPONENTS, THAT IS BY CALCULATING THE
C EIGENVECTORS, WE CAN DETERMINE NEW BASES FOR OUR COMPONENTS WHICH
C WILL BE INDEPENDENT.
C
      CALL EIGEN(DER,R,M,0)
      WRITE(6,220)
C
      DO 120 J=1,M
      WRITE(6,370)J,DER(J,J)
370   FORMAT(/,20X,'EIGENVALUE(',I2,') = ',E16.8,/)
      WRITE(6,380)(J,I,R(I,J),I=1,M)
380   FORMAT(25X,'EIGENVECTOR(',I2,',',I2,') = ',E16.8)
120   CONTINUE
C
      RETURN
      END
```

SUBROUTINE LISTING - COMYLD

```
C THIS SUBROUTINE DETERMINES THE ESTIMATES OF THE UNIT YIELDS.
C
      SUBROUTINE COMYLD(W,Q,M,N,CMPYLD)
      DIMENSION W(N),Q(N,M),CMPYLD(M)
C
      DO 10 I=1,N
      W(I)=0.0
      DO 15 J=1,M
      W(I)=W(I)+Q(I,J)*ALOG(CMPYLD(J))
15    CONTINUE
      W(I)=EXP(W(I))
10    CONTINUE
C
      RETURN
      END
```

SUBROUTINE LISTING - COMINF

```
C THIS SUBROUTINE CALCULATES THE INFORMATION MATRIX.
C
      SUBROUTINE COMINF(BB,M,N,XINF,W,CONDSD,LWORK,MWORK,CMPYLD)
      DIMENSION BB(N,M,M),XINF(M,M),W(N),LWORK(M),MWORK(M),
     $ CONDSD(M),CMPYLD(M)
C
      DO 10J=1,M
      DO 15K=1,J
      XINF(J,K)=0.0
      DO 20 I=1,N
      XINF(J,K)=XINF(J,K)+BB(I,J,K)*W(I)/(1.0-W(I))
20    CONTINUE
      XINF(J,K)=XINF(J,K)/(CMPYLD(J)*CMPYLD(K))
      XINF(K,J)=XINF(J,K)
      IF(J.EQ.K)CONDSD(J)=1.0/SQRT(XINF(J,J))
15    CONTINUE
10    CONTINUE
C
      CALL MINV(XINF,M,D,LWORK,MWORK)
C
      RETURN
      END
```

```
C THIS PROGRAM TAKES THE LOGARITHM OF THE UNIT YIELDS TO FORM A LINEAR
C EQUATION IN THE LOGARITHMS OF THE COMPONENT YIELDS.  THESE EQUATIONS
C ARE SOLVED USING MULTIPLE REGRESSION.  THESE ESTIMATES ARE THEN
C USED TO CONSTRUCT THE VERTICES OF A SIMPLEX WHICH WILL BE USED
C TO DETERMINE THE MAXIMUM LIKELIHOOD ESTIMATES OF THE COMPONENT
C YIELDS.
C
      DIMENSION Q(50,25),Y(50),YIELD(50),QTRANS(25,50),CMPYLD(25),
     $ L(25),W(50),CONS(25,1),MWORK(25),NEW(50),DER(25,25),Z(25,50),
     $ VERT(26,25),CENT(25),FLIKE(26),VERTR(25),VERTE(25),AHAT(25),
     $ VERTC(25),CONDSD(25),ADJYLD(50,1),QADJ(50,25),A(325),EST(50)
      REAL NEW
      READ(9,*) N,M
C
C N IS THE NUMBER OF ROWS (UNITS) & M IS THE NUMBER OF
C COLUMNS (COMPONENT FAMILIES).
      MPLUS=M+1
C
      CALL OLS(A,N,EST,MWORK,Q,YIELD,QTRANS,CMPYLD,L,M,NEW,
     $ DER,Z,CONS,Y,W,VERT,CENT,FLIKE,VERTR,VERTE,AHAT,VERTC,MPLUS,
     $ QADJ,ADJYLD,CONDSD)
      STOP
      END
C
      SUBROUTINE OLS(A,N,EST,MWORK,Q,YIELD,QTRANS,CMPYLD,L,M,NEW,
     $ DER,Z,CONS,Y,W,VERT,CENT,FLIKE,VERTR,VERTE,AHAT,VERTC,MPLUS,
     $ QADJ,ADJYLD,CONDSD)
      DIMENSION Q(N,M),VERT(MPLUS,M),CENT(M),FLIKE(MPLUS),YIELD(N),
     $ CMPYLD(M),EST(N),QTRANS(M,N),W(N),Y(N),L(M),MWORK(M),NEW(N),
     $ DER(M,M),VERTR(M),VERTE(M),AHAT(M),VERTC(M),Z(M,N),CONS(M,1),
     $ A(325),CONDSD(M),ADJYLD(N,1),QADJ(N,M)
      REAL NEW
C
      WRITE(6,200)
200   FORMAT(20X,'DATA MATRIX',/,/)
C
      DO 10 I=1,N
      READ(9,*)(Q(I,J),J=1,M),NEW(I),Y(I)
      WRITE(6,210)(Q(I,J),J=1,M)
210   FORMAT(3X,25F4.0)
10    CONTINUE
C
      WRITE(6,220)
220   FORMAT(1H1)
C
C CALCULATE THE YIELDS AND ADJUST FOR UNEQUAL VARIANCES.
C
      WRITE(6,230)
230   FORMAT(14X,'UNIT',13X,'UNITS TESTED',15X,'UNITS ACCEPTED',
     $ 14X,'UNIT YIELD',13X,'ADJUSTED YIELD',/)
C
      DO 15 I=1,N
      YIELD(I)=Y(I)/NEW(I)
      W(I)=SQRT((NEW(I)-Y(I))/(NEW(I)*Y(I)))
      ADJYLD(I,1)=ALOG(YIELD(I))/W(I)
```

PROGRAM LISTING - SIMPLEX (Cont'd)

```
      DO 20 J=1,M
      QADJ(I,J)=Q(I,J)/W(I)
20    CONTINUE
      WRITE(6,240)I,NEW(I),Y(I),YIELD(I),ADJYLD(I,1)
240   FORMAT(15X,I2,2(20X,F5.0),19X,F7.4,17X,F9.4)
15    CONTINUE
C
C DETERMINE THE TRANSPOSE OF THE ADJUSTED Q MATRIX.
C
      DO 25 I=1,N
      DO 30 J=1,M
      QTRANS(J,I)=QADJ(I,J)
30    CONTINUE
25    CONTINUE
C
C NOW WE CAN PERFORM THE ACTUAL REGRESSION.
C
      CALL GMPRD(QTRANS,QADJ,DER,M,N,M)
C
C THE MATRIX DER CONSISTS OF NON NEGATIVE ENTRIES.  HENCE WE CAN
C SCALE THE MATRIX SO THAT THE INVERSE OF A MATRIX WITH A SMALLER
C RANGE OF VALUES CAN BE DETERMINED.  THIS WILL REDUCE THE CHANCES
C OF THE PROGRAM CRASHING THROUGH A REGISTER OVERFLOW OR UNDERFLOW.
C
      ZZ=10000.*FLOAT(N)
C
      DO 35 J=1,M
      DO 40 JJ=1,M
      DER(J,JJ)=DER(J,JJ)/ZZ
40    CONTINUE
35    CONTINUE
C
C
C WE DETERMINE THE INVERSE OF DER USING THE SSP SUBROUTINE SINV
C WHICH DETERMINES THE INVERSE OF POSITIVE DEFINITE SYMMETRIC MATRICES.
C
      LSUM=0
C
      DO 45 J=1,M
      DO 50 K=1,J
      LSUM=LSUM+1
      A(LSUM)=DER(J,K)
50    CONTINUE
45    CONTINUE
C
      EPS=.001
      CALL SINV(A,M,EPS,IER)
      LSUM=0
C
      DO 55 J=1,M
      DO 60 K=1,J
      LSUM=LSUM+1
      DER(J,K)=A(LSUM)
      DER(K,J)=DER(J,K)
60    CONTINUE
```

PROGRAM LISTING - SIMPLEX (Cont'd)

```
55    CONTINUE
C
C NOW WE HAVE TO RESTORE DER TO ITS ORIGINAL SCALING BEFORE WE
C CAN USE IT.
C
      DO 65 J=1,M
      DO 70 JJ=1,M
      DER(J,JJ)=DER(J,JJ)/ZZ
70    CONTINUE
65    CONTINUE
C
      CALL GMPRD(DER,QTRANS,Z,M,M,N)
      CALL GMPRD(Z,ADJYLD,CONS,M,N,1)
C
      DO 75 J=1,M
      CMPYLD(J)=EXP(CONS(J,1))
75    CONTINUE
C
      WRITE(6,220)
      WRITE(6,250)
250   FORMAT(30X,'STARTING VALUES FOR THE SIMPLEX PROCEDURE:')
      WRITE(6,260)(I,CMPYLD(I),I=1,M)
260   FORMAT(15X,'COMPONENT YIELD(',I2,') = ',F15.7)
      ALPHA=1.0
      BETA=.5
      GAMMA=1.5
C
C DETERMINE THE VERTICES OF AN M DIMENSIONAL SIMPLEX WITH WHICH TO
C START THE SIMPLEX PROCEDURE.
C
      NCOUNT=0
      NSTOP=M*15
      WRITE(6,220)
C
      DO 80 J=1,M
      VERT(1,J)=CMPYLD(J)
80    CONTINUE
C
      DO 85 I=1,M
      DO 90 J=1,M
      VERT(I+1,J)=CMPYLD(J)
      IF(I.EQ.J)VERT(I+1,J)=CMPYLD(J)+.0005
90    CONTINUE
85    CONTINUE
C
      CALL XCALC(N,M,Q,MPLUS,VERT,FLIKE,AHAT,Y,NEW)
C
C AT THIS POINT WE CAN ACTUALLY BEGIN THE SIMPLEX PROCEDURE.
C
500   IF(NCOUNT.GE.NSTOP) GOTO 550
      CALL XCENT(M,MPLUS,VERT,CENT)
      NCOUNT=NCOUNT+1
C
C FIRST WE TRY A REFLECTION OF THE VERTEX WHICH GIVES A MINIMUM
C VALUE TO THE LIKELIHOOD FUNCTION.
```

PROGRAM LISTING - SIMPLEX (Cont'd)

```
      CALL XREF(M,MPLUS,ALPHA,VERT,CENT,VERTR)
      FLIKER=XF(N,M,Q,VERTR,Y,NEW)
1     IF(FLIKE(1).GE.FLIKER.AND.FLIKER.GE.FLIKE(M))GOTO 510
2     IF(FLIKER.GT.FLIKE(1)) GOTO 520
3     IF(FLIKE(M).GT.FLIKER.AND.FLIKER.GT.FLIKE(M+1)) GOTO 530
4     IF(FLIKE(M+1).GT.FLIKER) GOTO 540
      WRITE(6,270)
270   FORMAT(10X,'PROCEDURE HALTED')
      RETURN
C
C IF CONDITIONS OF STATEMENT 1 ARE SATISFIED, WE REPLACE THE VERTEX WITH
C MINIMUM LIKELIHOOD FUNCTION WITH THE REFLECTED POINT AND THEN RESTART
C THE PROCEDURE.
C
510   CALL XPLACE(M,MPLUS,VERT,FLIKE,VERTR,FLIKER)
      WRITE(6,280)NCOUNT,FLIKE(1)
280   FORMAT(10X,'NCOUNT =',I4,10X,'REFLECTION',9X,'LIKELIHOOD = ',
     $ E18.10)
      GOTO 500
C
C IF CONDITIONS OF STATEMENT 2 ARE SATISFIED WE TRY AN EXPANSION AND
C THEN RESTART THE PROCEDURE.
C
520   CALL XPAND(M,GAMMA,VERTE,CENT,VERTR)
      FLIKEE=XF(N,M,Q,VERTE,Y,NEW)
      IF(FLIKEE.GT.FLIKE(1))CALL XPLACE(M,MPLUS,VERT,FLIKE,VERTE,FLIKEE)
      IF(FLIKEE.LE.FLIKE(1))CALL XPLACE(M,MPLUS,VERT,FLIKE,VERTR,FLIKER)
      WRITE(6,290)NCOUNT,FLIKE(1)
290   FORMAT(10X,'NCOUNT =',I4,10X,'EXPANSION',10X,
     $ 'LIKELIHOOD = ',E18.10)
      GOTO 500
C
C IF CONDITIONS OF STATEMENT 3 ARE SATISFIED WE TRY A CONTRACTION.
C IF SUCCESSFUL WE RESTART THE PROCEDURE, OTHERWISE WE SHRINK THE
C SIMPLEX ABOUT THE POINT WITH THE HIGHEST LIKELIHOOD VALUE.
C
530   CALL XCONTR(M,MPLUS,BETA,VERT,VERTC,CENT)
      FLIKEC=XF( N,M,Q,VERTC,Y,NEW)
      IF(FLIKEC.GE.FLIKE(M+1)) GOTO 540
      IF(FLIKE(M+1).GT.FLIKEC)CALL XPLACE(M,MPLUS,VERT,FLIKE,VERTC,
     $ FLIKEC)
      WRITE(6,300)NCOUNT,FLIKE(1)
300   FORMAT(10X,'NCOUNT =',I4,10X,'CONTRACTION',10X,
     $ 'LIKELIHOOD = ',E18.10)
      GOTO 500
C
C IF CONTRACTION FAILS WE SHRINK THE SIMPLEX ABOUT THE VERTEX WHICH
C HAS THE GREATEST LIKELIHOOD VALUE.
C
540   CALL XSHR(M,MPLUS,VERT,FLIKE)
      CALL XCALC(N,M,Q,MPLUS,VERT,FLIKE,AHAT,Y,NEW)
      WRITE(6,310)NCOUNT,FLIKE(1)
310   FORMAT(10X,'NCOUNT =',I4,10X,'SHRINK',13X,'LIKELIHOOD = ',
     $ E18.10)
```

PROGRAM LISTING - SIMPLEX (Cont'd)

```
      GOTO 500
C
C DETERMINE THE UNIT YIELDS IN ORDER TO ESTIMATE THE COVARIANCE MATRIX.
C
550   LSUM=0.0
C
      DO 95 I=1,N
      W(I)=0.0
      DO 100 J=1,M
      W(I)=W(I)+Q(I,J)*ALOG(VERT(1,J))
100   CONTINUE
      W(I)=EXP(W(I))
95    CONTINUE
C
C ESTIMATE THE VARIANCE-COVARIANCE MATRIX.
C
      DO 105 J=1,M
      DO 110 K=1,J
      LSUM=LSUM+1
      A(LSUM)=0.0
C
      DO 115 I=1,N
      IF(Q(I,J).EQ.0.0)GOTO 115
      IF(Q(I,K).EQ.0.0)GOTO 115
      XNUM=NEW(I)*Q(I,K)*Q(I,J)*W(I)
      DEN=VERT(1,J)*VERT(1,K)*(1.0-W(I))
      A(LSUM)=A(LSUM)+XNUM/DEN
115   CONTINUE
C
      IF(K.EQ.J)CONDSD(J)=1.0/SQRT(A(LSUM))
      A(LSUM)=A(LSUM)/ZZ
110   CONTINUE
105   CONTINUE
C
      EPS=10.E-5
      CALL SINV(A,M,EPS,IER)
      LSUM=0
C
      DO 120 J=1,M
      DO 125 K=1,J
      LSUM=LSUM+1
      DER(J,K)=A(LSUM)/ZZ
      DER(K,J)=DER(J,K)
125   CONTINUE
120   CONTINUE
C
      WRITE(6,220)
      WRITE(6,320)
320   FORMAT(50X,'COVARIANCE MATRIX',/)
      WRITE(6,330)(J,J=1,M)
330   FORMAT(16(13X,I2),/)
C
      DO 130 J=1,M
      WRITE(6,340)J,(DER(I,J),I=1,J)
340   FORMAT(2X,I2,16(2X,E13.7))
```

PROGRAM LISTING - SIMPLEX (Cont'd)

```
130   CONTINUE
C
C DETERMINE APPROXIMATE CONFIDENCE INTERVALS FOR OUR COMPONENT
C FAMILY YIELD ESTIMATES.
C
      WRITE(6,220)
      WRITE(6,350)
350   FORMAT(25X,'APPROXIMATE 95% CONFIDENCE INTERVALS FOR',
     $ ' COMPONENT YIELDS:',/,/)
      WRITE(6,360)
360   FORMAT(7X,'COMPONENT',6X,'ESTIMATED',8X,'UNCONDITIONAL',
     $ 16X,'CONDITIONAL')
      WRITE(6,370)
370   FORMAT(8X,'FAMILY',10X,'YIELD',10X,'VARIANCE C.I.',
     $ 16X,'VARIANCE C.I.',/)
      WRITE(6,380)
380   FORMAT(10X,2('='),10X,8('='),6X,19('='),5X,23('='))
C
      DO 135 J=1,M
      COMPLB=VERT(1,J)-1.96*SQRT(DER(J,J))
      COMPUB=VERT(1,J)+1.96*SQRT(DER(J,J))
      CONDLB=VERT(1,J)-1.96*CONDSD(J)
      CONDUB=VERT(1,J)+1.96*CONDSD(J)
      WRITE(6,390)J,VERT(1,J),COMPLB,COMPUB,CONDLB,CONDUB
390   FORMAT(10X,I2,10X,F8.5,':',5X,'(',F8.5,',',F8.5,')',
     $ 5X,'(',F10.5,',',F10.5,')')
135   CONTINUE
C
C PRINT OUT THE ESTIMATED YIELD AND COMPARE WITH THAT OBSERVED.
C
      WRITE(6,220)
      WRITE(6,400)
400   FORMAT(9X,'UNIT',11X,'YIELD ESTIMATE',9X,
     $ 'OBSERVED YIELD',4X,'95% C.I. FOR OBSERVED YIELD',/)
      WRITE(6,410)
410   FORMAT(10X,2('='),2(15X,8('=')),15X,19('='))
C
      DO 140 I=1,N
      FAC=YIELD(I)*(1.0-YIELD(I))/NEW(I)
      FAC=1.96*SQRT(FAC)
      BINLB=YIELD(I)-FAC
      BINUB=YIELD(I)+FAC
      WRITE(6,420)I,W(I),YIELD(I),BINLB,BINUB
420   FORMAT(10X,I2,2(15X,F8.5),15X,'(',F8.5,',',F8.5,')')
140   CONTINUE
C
      RETURN
      END
```

SUBROUTINE LISTING - XCALC

```
      SUBROUTINE XCALC(N,M,Q,MPLUS,VERT,FLIKE,AHAT,Y,NEW)
C
C THIS SUBROUTINE CALCULATES THE VALUES OF THE LIKELIHOOD FUNCTION AT
C ALL OF THE VERTICES AND SORTS THE VERTICES IN ORDER OF SIZE OF THEIR
C LIKELIHOOD FUNCTIONS (LARGEST TO SMALLEST).
C
      DIMENSION VERT(MPLUS,M),Y(N),NEW(N),FLIKE(MPLUS),AHAT(M),Q(N,M)
      REAL NEW
C
      DO 10 I=1,MPLUS
      DO 15 J=1,M
      AHAT(J)=VERT(I,J)
15    CONTINUE
      FLIKE(I)=XF(N,M,Q,AHAT,Y,NEW)
10    CONTINUE
C
      DO 20 I=1,M
      IPLUS=I+1
      DO 25 II=IPLUS,MPLUS
      IF(FLIKE(I).GE.FLIKE(II)) GOTO 25
      DO 30 J=1,M
      TEMP=VERT(I,J)
      VERT(I,J)=VERT(II,J)
      VERT(II,J)=TEMP
30    CONTINUE
      TEMP=FLIKE(I)
      FLIKE(I)=FLIKE(II)
      FLIKE(II)=TEMP
25    CONTINUE
20    CONTINUE
C
      RETURN
      END
```

SUBROUTINE LISTING - XPLACE

```
      SUBROUTINE XPLACE(M,MPLUS,VERT,FLIKE,VERTR,FLIK)
C
C THIS SUBROUTINE PLACES OUR NEW VERTEX IN THE PROPER ORDER AMONGST THE
C OTHER VERTICES
C
      DIMENSION VERT(MPLUS,M),FLIKE(MPLUS),VERTR(M)
C
C REPLACE THE VERTEX WITH THE LEAST LIKELIHOOD BY OUR NEWLY CHOSEN
C VERTEX.
C
      FLIKE(MPLUS)=FLIK
      DO 10 J=1,M
      VERT(MPLUS,J)=VERTR(J)
10    CONTINUE
C
      DO 15 I=1,M
      IF(FLIKE(MPLUS-I).GE.FLIKE(MPLUS-I))GOTO 500
      DO 20 J=1,M
      TEMP=VERT(MPLUS-I,J)
      VERT(MPLUS-I,J)=VERT(MPLUS-I+1,J)
      VERT(MPLUS-I+1,J)=TEMP
20    CONTINUE
      TEMP=FLIKE(MPLUS-I)
      FLIKE(MPLUS-I)=FLIKE(MPLUS-I+1)
      FLIKE(MPLUS-I+1)=TERM
15    CONTINUE
C
500    RETURN
      END
```

SUBROUTINE LISTING - XREF

```
      SUBROUTINE XREF(M,MPLUS,ALPHA,VERT,CENT,VERTR)
C
C THIS SUBROUTINE FINDS A POINT REFLECTED THROUGH THE CENTROID
C FROM THE VERTEX AT WHICH THE FUNCTION IS MINIMUM.
C
      DIMENSION VERT(MPLUS,M),CENT(M),VERTR(M)
      DO 10 J=1,M
      VERTR(J)=(1.0+ALPHA)*CENT(J)-ALPHA*VERT(MPLUS,J)
10    CONTINUE
C
      RETURN
      END
```

FUNCTION LISTING - XF

```
      FUNCTION XF(N,M,Q,AHAT,Y,NEW)
C
C THIS FUNCTION SUBPROGRAM DETERMINES THE LOGARITHM OF THE
C LIKELIHOOD FUNCTION AT A PARTICULAR VERTEX.  THE LIKELIHOOD
C FUNCTION WILL HAVE A MAXIMUM AT THE SAME POINT AS THE LOG
C OF THE LIKELIHOOD FUNCTION.
C
      DIMENSION Q(N,M),NEW(N),Y(N),AHAT(M)
      REAL NEW
      XF=0.0
      DO 15 I=1,N
      XK=0.0
      DO 25 J=1,M
      XK=XK+Q(I,J)*ALOG(AHAT(J))
25    CONTINUE
      IF(XK.GE.0.0)GOTO 35
      XK=Y(I)*XK+(NEW(I)-Y(I))*ALOG(1.0-EXP(XK))
      XF=XF+XK
15    CONTINUE
      RETURN
35    XF=16.E-65
RETURN
      END
```

SUBROUTINE LISTING - XCONTR

```
      SUBROUTINE XCONTR(M,MPLUS,BETA,VERT,VERTC,CENT)
C
C THIS SUBROUTINE DECREASES THE SIZE OF THE SIMPLEX BY MOVING THE
C REFLECTED VERTEX IN.
C
      DIMENSION VERT(MPLUS,M),VERTC(M),CENT(M)
C
      DO 10 J=1,M
      VERTC(J)=BETA*VERT(MPLUS,J)+(1.0-BETA)*CENT(J)
10    CONTINUE
C
      RETURN
      END
```

SUBROUTINE LISTING - XCENT

```
      SUBROUTINE XCENT(M,MPLUS,VERT,CENT)
C THIS SUBROUTINE LOCATES THE CENTROID OF THE SIMPLEX EXCLUDING THE
C VERTEX HAVING THE LEAST LIKELIHOOD.
C
      DIMENSION VERT(MPLUS,M),CENT(M)
      DO 15 I=1,M
      CENT(I)=0.0
      DO 25 J=1,M
      CENT(I)=CENT(I)+VERT(J,I)
25    CONTINUE
      CENT(I)=CENT(I)/FLOAT(M)
15    CONTINUE
      RETURN
      END
```

SUBROUTINE LISTING - XSHR

```
      SUBROUTINE XSHR(M,MPLUS,VERT,FLIKE)
C
C THIS SUBROUTINE SHRINKS THE SIMPLEX,RETAINING THE VERTEX WITH THE
C MAXIMUM LIKELIHOOD.
C
      DIMENSION VERT(MPLUS,M),FLIKE(MPLUS)
      DO 15 I=1,M
      DO 25 J=1,M
      VERT(I+1,J)=(VERT(I+1,J)+VERT(1,J))/2.0
25    CONTINUE
15    CONTINUE
      RETURN
      END
```

SUBROUTINE LISTING - XPAND

```
      SUBROUTINE XPAND(M,GAMMA,VERTE,CENT,VERTR)
C
C THIS SUBROUTINE EXPANDS THE REFLECTION IN THE DIRECTION ALONG WHICH
C A FURTHER IMPROVEMENT OF THE LIKELIHOOD VALUE IS EXPECTED
C
      DIMENSION VERTE(M),CENT(M),VERTR(M)
C
      DO 10 J=1,M
      VERTE(J)=GAMMA*VERTR(J)+(1.0-GAMMA)*CENT(J)
10    CONTINUE
C
      RETURN
      END
```

REFERENCES

Anderson, T.W. Introduction to Multivariate Statistical Analysis. John Wiley and Sons Inc., 1958.

Kendall, M.G. and A. Stuart. The Advanced Theory of Statistics. Volume 1. Hafner Publishing Company, 1969.

Kendall, M.G. and A. Stuart. The Advanced Theory of Statistics. Volume 2. Hafner Publishing Company, 1973.

Kowalik, J. and M.R. Osborne. Methods for Unconstrained Optimization Problems. American Elsevier Publishing Company, Inc., 1968.

Myers, R., K. Wong, and H. Gordy. Reliability Engineering for Electronic Systems. John Wiley and Sons Inc., 1964.

Rao, C.R. Linear Statistical Inference and Its Applications. John Wiley and Sons Inc., 1965.

Russell, D.L. Optimization Theory. W.A. Benjamin, Inc., 1970.