MULTIVARIATE NON-NORMAL PROCESS CAPABILITY INDICES: A SIMULATION APPROACH

By

ALLEN L. LEWIS

Bachelor of Science

Iowa State University

Ames, Iowa

1992

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 1998

MULTIVARIATE NON-NORMAL PROCESS

CAPABILITY INDICES: A SIMULATION

APPROACH

Thesis Approved:

Kenneth E Case

Thesis Adviser

David B. Pratt

Camille F. DeYong

Wayne B. Powell

Dean of the Graduate College

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my major advisor, Dr. Kenneth E. Case, for his guidance, support, friendship, and eternal patience through my graduate academic experience and in completion of this study. I also wish to thank my other committee members, Dr. David Pratt and Dr. Camille DeYong, whose constructive comments and support were essential in the completion of this study. I also thank Dr. David W. Miller and Mr. K. B. Williams for sharing expertise.

Without the financial assistance provided by both Appleton Electric Company in Columbus, Nebraska and MerCruiser in Stillwater, Oklahoma, I may not have been able to complete my graduate education or this study. A special thanks to my manager, Mr. Tony Rudd, Director of Quality at MerCruiser, for his continued support of my academic career, patience with my class schedules, and friendship.

I cannot possibly begin to thank my wonderful wife, Carolyn, for the years of patience, encouragement, love, and understanding that she has given me from the start. Thanks to my two sons, Nicholas and Carston, for their understanding and willingness to allow their father to study, with minimal interruption, in the computer room. I would also like to take this opportunity to thank my parents, Gerald and Lorraine, my wife's parents, Don and Mary Lou, my brother, Jason, and my sister-in-law, Jennifer, for their support and encouragement. The support of other family members and friends was also greatly appreciated.

Finally, I would like to thank the Department of Industrial Engineering and Management at Oklahoma State University for providing an outstanding curriculum and staff. They have truly allowed me to learn.

TABLE OF CONTENTS

Chapter		Page

LIST OF TABLES

Table		Page

CHAPTER I

INTRODUCTION

1.1 Introduction

In modern industry, process capability drives capital investment decisions, as well as quality improvement programs. Processes are usually comprised of many variables, a few of which are usually either very important to the success of the process or have some safety implication. Many times these variables are both correlated, Wierda (1993), and non-normally distributed, Western Electric (1956) and Rivera (1995), and thus produce inaccurate and deceiving capability metrics when existing univariate methods are employed. Therefore, a serious need exists to determine a capability index which accurately describes the ability of a multivariate, correlated, and non-normal process to produce product within known specification limits.

1.2 Background

1.2.1 Univariate Process Capability Indices

A process capability index is a statistical measure of process performance. Sullivan (1984 & 1985) and Kane (1986a & 1986b) are two very well known references on the subject of process capability indices. They introduce and discuss the two most

CHAPTER I

INTRODUCTION

1.1 Introduction

In modern industry, process capability drives capital investment decisions, as well as quality improvement programs. Processes are usually comprised of many variables, a few of which are usually either very important to the success of the process or have some safety implication. Many times these variables are both correlated, Wierda (1993), and non-normally distributed, Western Electric (1956) and Rivera (1995), and thus produce inaccurate and deceiving capability metrics when existing univariate methods are employed. Therefore, a serious need exists to determine a capability index which accurately describes the ability of a multivariate, correlated, and non-normal process to produce product within known specification limits.

1.2 Background

1.2.1 Univariate Process Capability Indices

A process capability index is a statistical measure of process performance. Sullivan (1984 & 1985) and Kane (1986a & 1986b) are two very well known references on the subject of process capability indices. They introduce and discuss the two most industry-recognized indices, C_p and C_{pk}. The C_p index is defined as the ratio of allowable process spread to actual process spread, under the normal distribution assumption, and is calculated as follows:

$$C_p = \frac{\text{allowable process spread}}{\text{actual process spread}} = \frac{USL - LSL}{6\sigma} = \frac{USL - LSL}{NT}$$

where USL and LSL represent the upper and lower specification limits, respectively, and sigma, σ, represents the statistical standard deviation of the process. Natural tolerance (NT) is defined as six process standard deviations and represents the actual process spread. Therefore, when the natural tolerance of the process is equal to the specified tolerance, the value of C_p is one. Figure 1.1 and Figure 1.2 graphically present this situation.

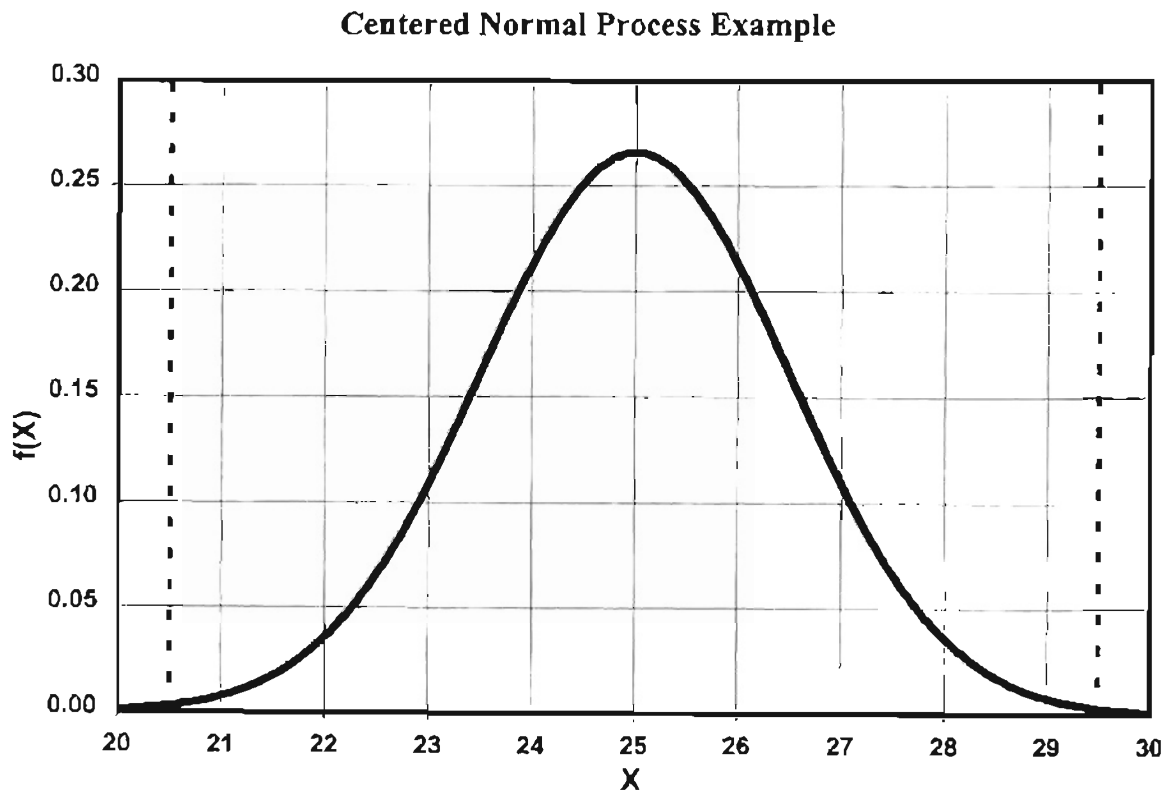


Figure 1.1: Centered Normal Process with $C_p = 1.0$

Figure 1.1 presents a process which is normally distributed with a mean, μ, of 25 and a standard deviation, σ, of 1.5. A conventional method of stating this is to say that the process is $\sim N(25,1.5^2)$. Notice that the USL is equal to 29.5 and the LSL is equal to 20.5. Therefore, the allowable process spread is 9.0 and the actual process spread is 9.0, which makes the ratio equal to 1.0. Figure 1.2 is presented to show one of the inherent problems with the C_p index.

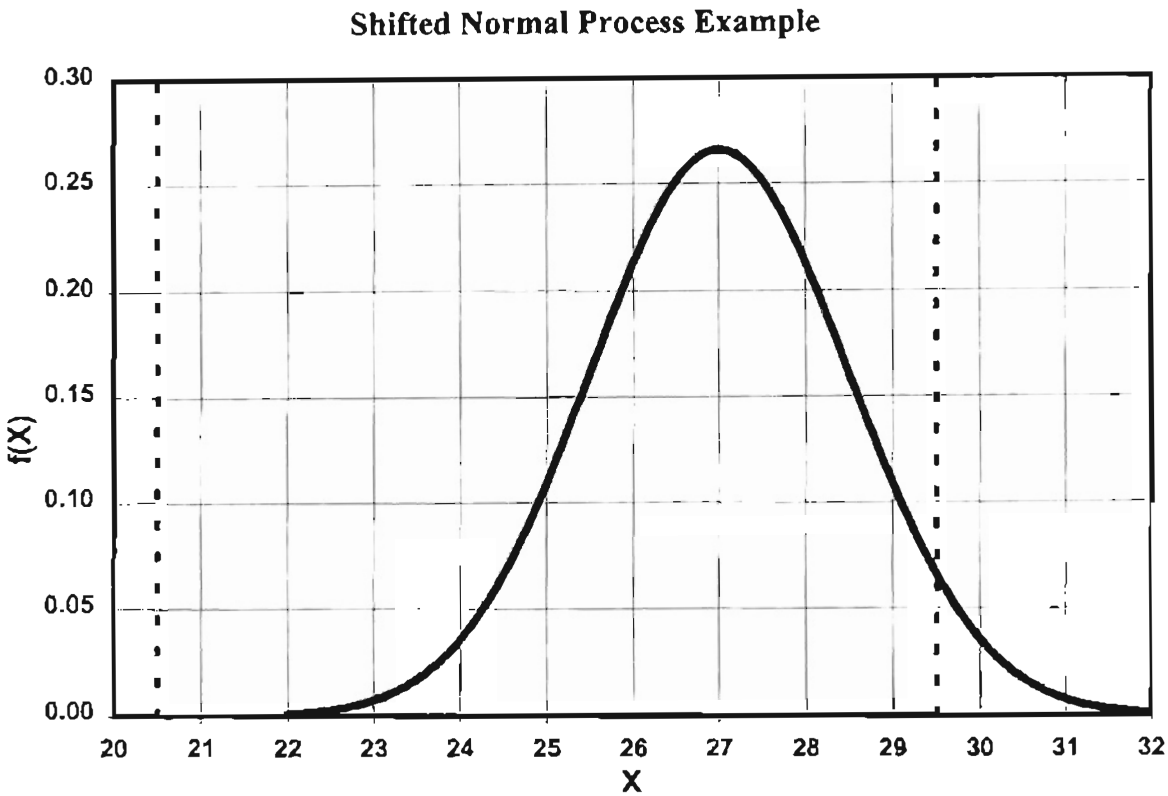


Figure 1.2: Shifted Normal Process with C_p = 1.0

Figure 1.2 presents the same process after the mean has shifted to 27.0, i.e., the process is $\sim N(27,1.5^2)$. Since the specification limits have not changed, the allowable process spread is still equal to 9.0 and because the process variation remained constant, the actual process spread is still equal to 9.0. Therefore, the value of C_p is still equal to 1.0, yet, it is easily seen that the shifted process will produce more product over the upper specification limit. This problem has been corrected with the additional process capability index, C_{pk}. The C_{pk} index has been defined in two different, but algebraically equivalent, ways. The first and most widely used definition, utilizes the upper, CPU, and lower, CPL, capability indices, and is calculated as follows:

$$C_{pk} = \mathrm{Min}(\mathrm{CPL}, \mathrm{CPU})$$

where

$$\mathrm{CPU} = \frac{\text{allowable upper spread}}{\text{actual upper spread}} = \frac{\mathrm{USL} - \mu}{3\sigma}$$

and

$$\mathrm{CPL} = \frac{\text{allowable lower spread}}{\text{actual lower spread}} = \frac{\mu - \mathrm{LSL}}{3\sigma}$$

The second definition utilizes the k-factor, which essentially is a correction factor or index for non-centered processes, and is defined as follows:

$$C_{pk} = C_p(1-k) \qquad \text{where} \qquad k = \frac{|T-\mu|}{\frac{\mathrm{USL}-\mathrm{LSL}}{2}}$$

The process target (T) is normally assumed to be half way in between the upper and lower specification limits, however, this is not always the situation found in industry. Therefore the k-factor equation is presented in generality and can be used when the target value is not centered. It is my experience that most processes are targeted in the middle of the specification limits simply because it minimizes the amount of product which is produced outside of the specification limits, with the normal distribution assumption. This seems to be the case even when the target has been specified as some other value.

If we apply these equations to the process presented on Figure 1.2, we calculate CPL = 1.44, CPU = 0.56, and k = 0.44. The value of C_{pk}, using either definition, is calculated as 0.56. Therefore, the C_{pk} index can be thought of as assigning a penalty to the process for being off-center. Note that the value of C_{pk} is equal to 1.0 when the equations are applied to the centered process presented in Figure 1.1. The C_p and C_{pk} indices are usually used together because C_p represents the potential process performance, if the process were centered, and C_{pk} represents the process performance with a correction factor for centering. Another way of stating this is that as the process moves towards the specification center, the value of C_{pk} approaches the value of C_p. In other words, the value of C_p is the best performance which the process could have without reducing the variation which exists in the process. Note that there is also a relationship between the upper and lower capability indices and the value of C_p:

$$C_p = \frac{\text{CPU} + \text{CPL}}{2}$$

Under the normal distribution assumption, we can also calculate the amount of product which is being produced outside of the specification limits. This amount is called the proportion nonconforming, p, and is presented in Littig et al. (1992) as follows:

$$p = 2 - \Phi(3\ C_p\ (1 - k)) - \Phi(3\ C_p\ (1 + k))$$

where $\Phi(z)$ is the cumulative standardized normal distribution, and can be found tabulated in most statistical, mathematical, and science texts. Using this equation we can state that when $C_{pk} = 1.0$ then p = 0.27% or 2,700 parts per million (ppm), when $C_{pk} =$ 1.33 then p = 64 ppm, and when $C_{pk} = 2.0$ then $p \leq 0.1$ ppm. The proportion nonconforming for the process presented in Figure 1.2 is 4.78%. It is this fundamental relationship between proportion nonconforming and these process capability indices which allows them to be as popular in industry as they currently are.

Many univariate process capability indices can be found in the literature, however, Chan et al. (1988a) and Boyles (1991a & 1991b) present a third index, C_{pm}, which utilizes a slightly different approach. The basic idea is that as the process average drifts further away from the target value, the penalty should increase. This approach is often called the loss function approach, where the loss function describes the rate at which the penalty or "loss" increases with the distance from target. Chan et al. (1988a) present the following definition:

$$C_{pm} = \frac{USL - LSL}{6\sigma'} \quad \text{where} \quad \sigma' = \sqrt{\mathbf{E}[\mathbf{X} - \mathbf{T}]^2}$$

Boyles (1991a) calls this index the Taguchi capability index because the loss function which was used had been presented in Taguchi (1985). Boyles presents the following similar definition:

$$C_{pm} = \frac{USL - LSL}{6\tau} \quad \text{where} \quad \tau^2 = \mathbf{E}\left[(\mathbf{X} - \mathbf{T})^2\right]$$

where τ is the standard deviation from target, and can easily be computed from the following equation:

$$\tau^2 = \sigma^2 + (\mu - T)^2$$

The C_{pm} index is not widely used in industry primarily because, in my opinion, it does not have the same fundamental relationship with the proportion nonconforming. This relationship is often referred to as "feel" or physical meaning. For example the C_{pm} value for the process of Figure 1.2 would be 0.6. Note that this value is relatively close to the calculated value of C_{pk}, however, if the process were moved further away from the target, the differences between the two indices would increase.

1.2.2 Univariate Non-Normal Processes

All of the indices presented in the previous section and most of those presented in the literature are based on the normal distribution assumption. Some authors state that the process capability indices are robust enough to give accurate results for "near-normal" distributions. However, it is my experience in industry that there are many distributions which cannot be considered either normal or near-normal. Other authors like Kane (1986a) suggest that the indices are very sensitive to normality and suggest the use of transformations or fitting the data with other distributions which have known properties. Clements (1989) presents the use of Pearson curves in conjunction with tabulated standardized tails of Pearson curves. Chan et al. (1988b) present a graphical technique for estimating distribution-free capability indices. The end result is that non-normal distributions are found in industry, and other techniques are required when assessing the capability of non-normal processes.

Figure 1.3 demonstrates a very simple example of what occurs when the normal distribution assumption is applied to a non-normal process. If an actual process followed the lognormal distribution displayed on Figure 1.3, then under the normal distribution assumption, the calculated mean and standard deviation would lead us to believe that the process was distributed as the normal distribution, also displayed on Figure 1.3. All process inferences, therefore, would be based upon this normal distribution. Note that the Lognormal distribution was not selected based on how common it is found in industry, but rather on how closely its shape resembled the normal distribution.

Univariate Lognormal Process Example

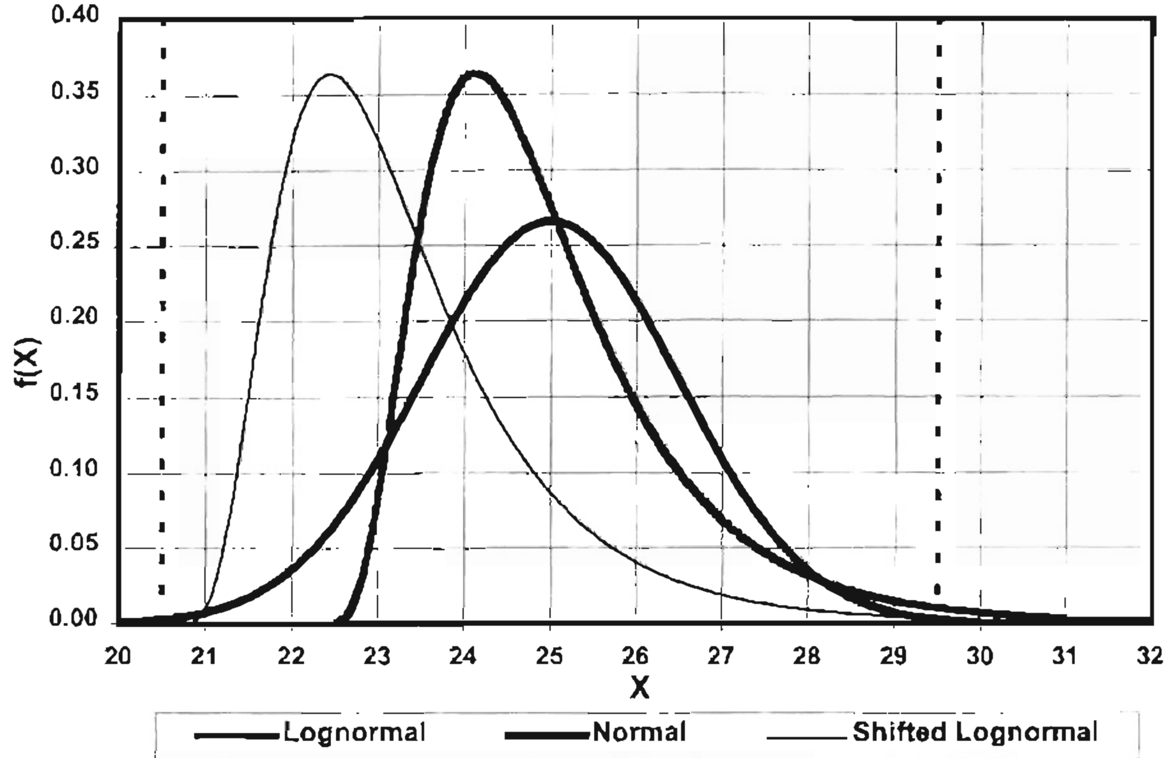


Figure 1.3: Lognormal Process under Normal Assumption

Also note that the example was created so that the resulting normal distribution is the same as the process displayed in Figure 1.1. Therefore, under the normal distribution assumption, the C_p and C_{pk} indices would both have the calculated value of 1.0. This would imply that the distribution was centered and therefore, had its proportion nonconforming minimized. It can be easily seen that this is definitely not the case, even for this lognormal distribution, which can appear approximately normal for small sample sizes. Figure 1.3 also displays a shifted lognormal which represents the required position of the process to minimize the proportion nonconforming. Note that the "calculated" normal distribution (not displayed) resulting from the shifted lognormal would yield lower capability indices and a higher proportion nonconforming. This example clearly demonstrates that non-normal distributions need to be handled differently, and that

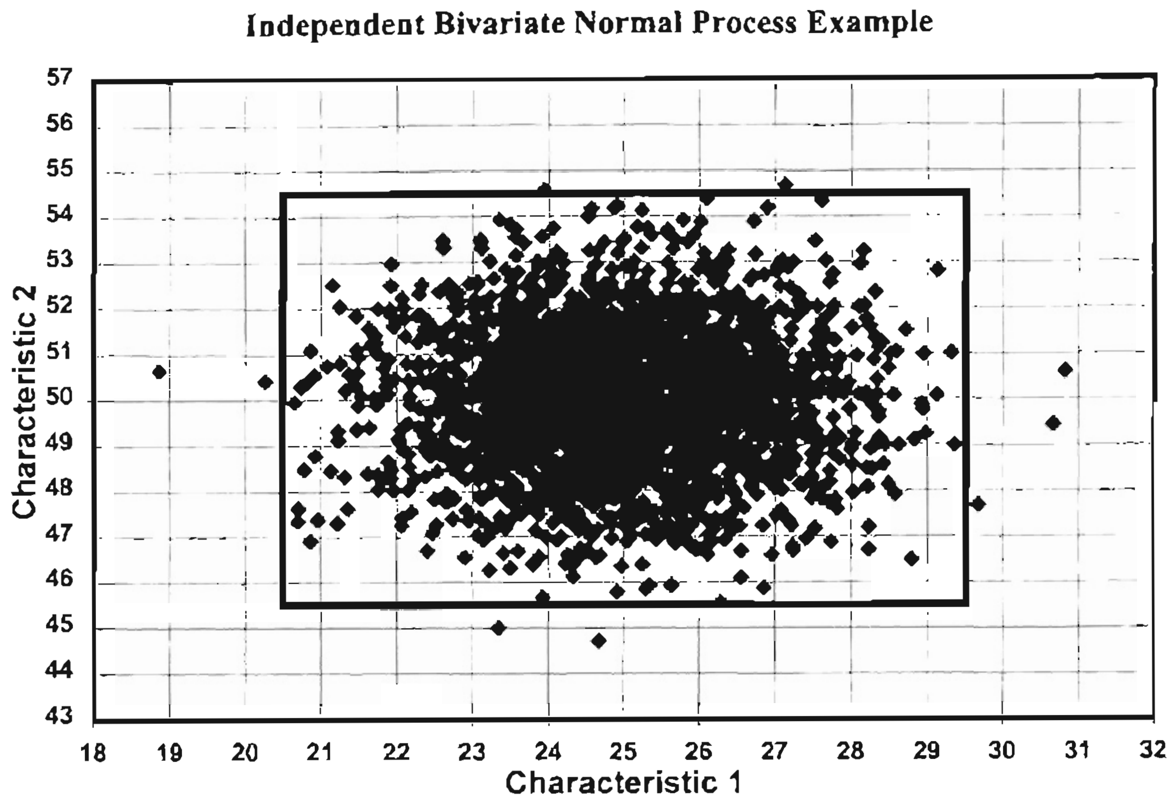


Figure 1.4: Independent Bivariate Normal Process

Some authors like Wierda (1993) have suggested that the indices for independent characteristics can be multiplied together to calculate the multivariate index. Although, at first glance, the idea seems somewhat intuitive, a simple example can dispel the logic. Consider 10 characteristics each with a C_{pk} value of 1.01. The resulting multivariate index would be 1.105. In other words, the capability of the process appears to be better than any of its individual characteristics. This capability synergism seems to be improbable, especially when we consider that if any one of the product's characteristics are outside of its individual specifications, then the entire product is considered to be nonconforming. This example leads us to the next problem, which is the calculation of the proportion nonconforming of the process. Upon inspection of the centered bivariate normal process displayed in Figure 1.4, it appears unlikely that a nonconforming point would be simultaneously outside of both specifications. It would seem logical, therefore, to assume that the proportion nonconforming for both characteristics together would simply be the sum of the proportions nonconforming for each characteristic. In this example where both characteristics each have 2,700 ppm nonconforming, the total nonconforming would be 5,400 ppm. If this were true, then it would also be logical to assume that the process capability index would decrease.

Unfortunately, processes found in industry are rarely as "perfect" as the example presented here. Processes can have dependent characteristics which are both non-normally distributed and not centered. Some of these situations are briefly discussed in the following sections.

1.2.4 Dependent Multivariate Processes

In the previous section, we discussed multiple characteristics which were considered independent or non-correlated. In other words, the value of any particular characteristic has absolutely zero effect on any of the other characteristics. While independent characteristics are certainly possible, many times when the characteristics come from the same process, the characteristics are dependent or correlated. In other words, the values of some characteristics either affect or are related to the values of others. Sometimes it is easy to think of this situation when you consider physical material properties like hardness and strength, which are obviously related. In order to demonstrate the effects of correlation, again we will simulate a centered bivariate normal distribution with each characteristic distributed the same as those in Figure 1.4. This time we will simulate dependence between the first characteristic and the second characteristic. The correlation coefficient, ρ, between them was set at 0.6. The distribution example is displayed on Figure 1.5.

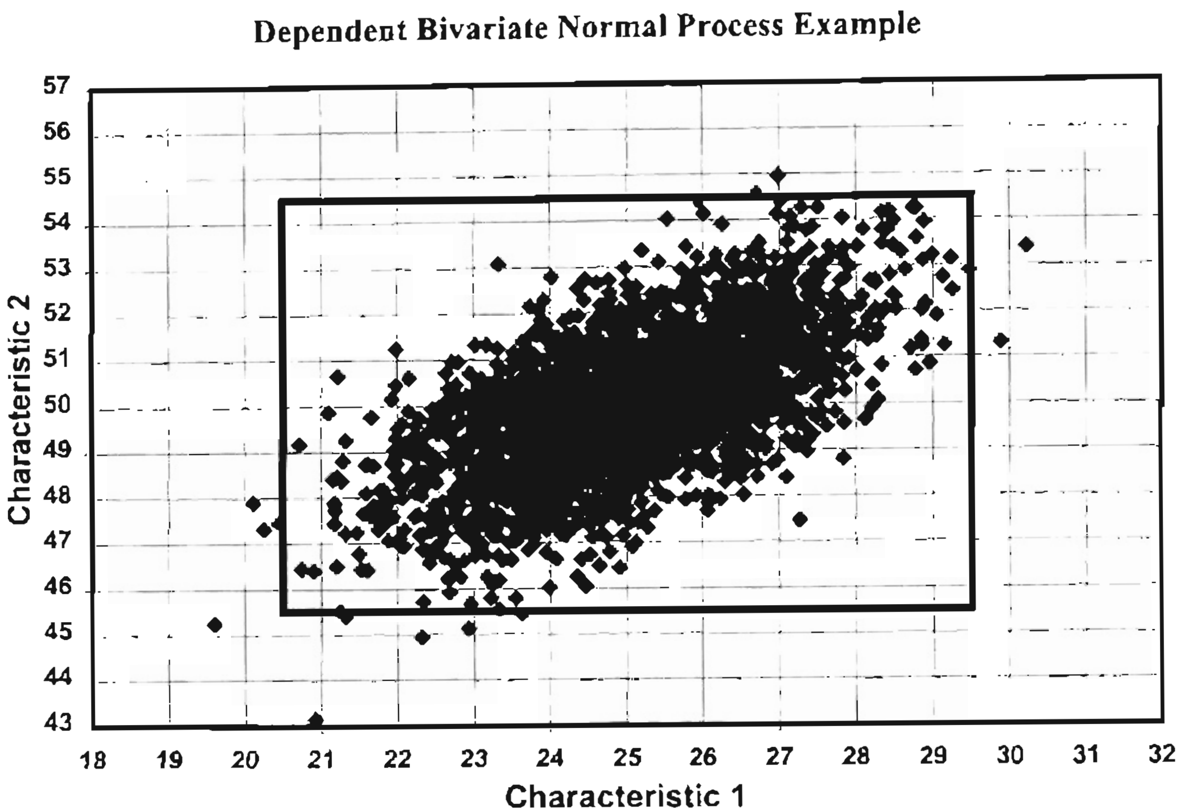


Figure 1.5: Dependent Bivariate Normal Process

There are a number of observations which can be made when comparing the correlated process of Figure 1.5 to the non-correlated process of Figure 1.4. The first observation which is usually made is that the shape and orientation of the process has changed in relationship to the specification limits. The most interesting result of this shape change is that it does not affect the capability index values calculated on either characteristic. Since the indices are simple functions of the characteristic's mean, standard deviation, and specifications, which the correlation did not affect, the indices do not change either. That is to say that C_p and C_{pk} have calculated values of 1.0 for both characteristics in Figure 1.5. One of the approaches found in the literature is to compare the area of the process ellipse to the area of the specification rectangle, where the process ellipse can be defined as an ellipse around the process which captures a specified probability contour of the process. If this approach were used, it would seem that the correlation would cause changes in the area of the process ellipse. However, as in most cases, the process ellipse is primarily a function of the multivariate normal assumption.

Another observation which can be made is that while it doesn't appear that the total proportion nonconforming has changed, it does appear that the probability that a given nonconforming product will have multiple characteristics nonconforming, has increased. In other words, while it was stated that the process in Figure 1.4 probably has about 5,400 ppm nonconforming, the process in Figure 1.5 probably has some proportion less than 5,400 ppm. This is because some of the 5,400 parts would be counted twice, if they were outside of the specifications for both characteristics. It would also seem to be a fair statement that this probability of being nonconforming on multiple specifications, would increase as the first characteristic's distribution moved away from its specified target. A final observation would be that if it were desired to change a correlated process to reduce its variation and one of the characteristics tended to be a controlling dimenesion, the attention would have to focus on the "controlling" characteristic. It would be the "controlling" characteristic that governs the apparent variation which is seen in the other dependent and "non-controlling" characteristic(s). This is an important observation for process engineers in industry.

1.2.5 Summary

The previous sections have discussed the effects of non-normality, multiple characteristics, characteristic correlation, and their combined effects upon the estimation of process capability indices. Attempts have been made to address some of these issues, however, the issue of multivariate non-normality has been virtually ignored in the literature. This is very unfortunate because the situation can be encountered every day in industry. A thorough review of the existing literature can be found in the next chapter.

1.3 Statement of the Problem

Although the univariate process capability indices, C_p and C_{pk}, have been used for many years to describe the performance of individual characteristics of a process, no practical approach has been presented for the calculation of process capability indices when applied to either univariate non-normal processes or multivariate non-normal processes.

1.4 Research Objective

This research proposes to create a new methodology for assessing the capability of a non-normal, multivariate, correlated process, and to develop a multivariate process capability index that will ensure a meaningful physical interpretation of the process capability. Accomplishing the objective requires several subobjectives. These are:

1. Evaluate the univariate marginal distributions and determine a probability function which adequately describes each distribution.

2. Evaluate the correlation between the univariate marginal distributions.
3. Simulate the multivariate distribution based upon the correlation and parameters of the marginal distributions.
4. Calculate the proportion of nonconforming product for the multivariate process, as well as for each of the marginal distributions.
5. Transform the multivariate proportion nonconforming into a capability index.

The Johnson system of distributions was chosen to describe all univariate marginal distributions, primarily due to the system's ability to "fit" most realistic distribution shapes. Another primary reason for choosing the Johnson system is that the Johnson equations all represent transformations to the standard normal distribution. Convenient relationships between the different distributions of the Johnson system can be derived based on this transformation result, which allow the simulation of a multivariate Johnson distribution to be possible. The Johnson system of distributions will be discussed at length in Chapter 3.

1.5 Delimitations

There are several important delimitations of this research. These are:

1. The study will address multivariate processes with a maximum of four marginal distributions.
2. The study will only use the Johnson distribution family of probability functions to describe the marginal distribution data.
3. The study will not address dependent specifications.

4. The study will only address the statistical properties of the proposed capability index through simulation techniques.

1.6 Assumptions

There are several important assumptions of this research. These are:

1. The need for understanding the capability of multivariate processes will continue.
2. All marginal distributions encountered in industry can be adequately described using the Johnson distribution family of probability functions.
3. All parameters of the Johnson probability functions may not require optimization to ensure an adequate fit to the marginal distributions.
4. When multivariate data is presented, the first marginal distribution will be treated as independent by the sample generator.
5. When all marginal distributions are adequately described by the normal distribution, then the process will be considered multivariate normally distributed.
6. The original multivariate distribution can be recreated with knowledge of the marginal distributions and their correlation.
7. If the situation is encountered where the strategy to minimize nonconforming product competes with the strategy to minimize the deviation from target, then the minimization of nonconforming product will take priority.
8. All processes which are evaluated are assumed to be in statistical control.

1.7 Contributions

This research becomes the first step to provide a new methodology in assessing the performance of a multivariate process, independent of the distribution's shape. The contributions of the research are as follows:

1. Provide an estimate of the proportion of product which is nonconforming to its specification limits for a multivariate process containing up to four marginal distributions, independent of their shape.
2. Provide a multivariate process capability index which is related to the estimate of proportion nonconforming, to ensure a similar physical interpretation as with the current univariate process capability indices, C_p and C_{pk}.
3. Provide as software application which can be easily utilized by practicing process engineers.
4. Provide a starting point for further advanced study which can include dependent specifications, combinations of dependent and independent specifications, more than four variables, and additional potential probability distribution systems to more adequately fit all distributions which can be encountered in industry.

CHAPTER II

LITERATURE REVIEW

2.1 Process Capability Indices

Since process capability indices were introduced to industry as a measure of process performance, Sullivan (1984 & 1985) and Kane (1986a & 1986b), there have been large amounts of literature published which sufficiently cover the subject. The definitions of C_p and C_{pk} are common knowledge and widely used in industry today. It did not take long before limitations were discovered with these indices, especially as they apply to non-normal processes and correlated multivariate processes. These limitations are not widely known in industry, and the univariate indices continue to be blindly misused as an indicator of process capability. As a practicing quality engineer, I have found it very difficult to adequately convey the capability of a process or the capability of equipment, when they consist of many variables, and therefore, many capability estimates. A thorough review of the literature has produced a number of multivariate process capability measures.

Wang et al. (1996) lists a majority of the current literature on multivariate capability, and groups the approaches into three categories: 1) Vector, matrix, or loss function approaches, 2) Tolerance region to process region ratio approaches, and 3)

Probability of nonconforming product approaches. I believe that these are very logical categories, and I will utilize them to review the current literature.

2.2 Vector, Matrix, and Loss Function Approaches

Hubele, Shahriari and Cheng (1991) propose a three component capability vector (CV) which is based on the bivariate normal distribution with independent (rectangular) specifications. This component is a bivariate analogue of the univariate Cp. The first component represents the ratio of the specification rectangle area to the process rectangle (defined as the smallest rectangle which circumscribes a given probability ellipse of the process) area. The second component is defined as the significance level of Hotelling's T^2 statistic. This component represents the distance between the specification centers and the process average. The third component indicates whether the process rectangle is completely contained within the specification rectangle. Wang et al. (1996) extends this vector to the general multivariate case based on the multivariate normal distribution, and also conveniently names the vector and its components: MPCV = [C_{pM}, PV, LI]. The third component, LI, was changed to a simple binary index, probably due to the lack of physical interpretation of the exact value as presented in the Hubele et al. (1991) paper. It should be noted that specification and process areas in the bivariate case are referred to as specification and process volumes in the general multivariate case.

Kotz and Johnson (1993) propose two indices based on the multivariate normal distribution: E[L(X)] and ${}_vC_R$. E[L(X)] is the expected value of a quadratic loss function, L(X), expressed in terms of the specified matrix, A. The ${}_vC_R$ index consists of the loss function value added to the determinant ratio of the variance – covariance matrix to the A matrix.

2.3 Tolerance Region to Process Region Ratio Approaches

Chan, Cheng and Spiring (1988a & 1991) propose the multivariate process capability measure, $\underset{\sim}{C}pm$, and its associated properties, based on the multivariate normal distribution and ellipsoidal specifications. The proposed index is the square root ratio of degrees of freedom over the sum of the observed Mahalanobis distances. The index is compared to the W test statistic under a one-sided hypothesis of capability. The expected value and variance of the index are also derived.

The first component of the capability vector proposed by Hubele et al. (1991), CpM (discussed earlier), can be considered as a single index falling into this category.

Kocherlakota and Kocherlakota (1991) propose the bivariate joint distribution of two standard univariate Cp's which are based on the univariate and bivariate normal distributions and utilize rectangular specifications.

Pearn, Kotz and Johnson (1992) propose two indices, ${}_vC_p$ and ${}_vC_{pm}$, which are extensions of the index proposed by Chan et al. (1988a & 1991) based on the multivariate normal distribution and ellipsoidal specifications.

Taam, Subbaiah and Liddy (1993) propose the multivariate capability index, MC_{pm}, which, in its base form is not necessarily based on the multivariate normal distribution. It is simply presented as the volume ratio of the modified tolerance region to the scaled process region. However, calculating these volumes in the non-normal case, can be its own challenge. A shape correction factor is introduced to handle non-elliptical specifications, and the proposed index is manipulated to provide both a process variation component and a distance from target component. The multivariate normal and elliptical specification assumptions seem to be the basis of the approach. A bivariate normal example is presented.

2.4 Probability of Nonconforming Product Approaches

Littig, Lam and Pollock (1992) propose two multivariate capability indices, C_{p^*} and C_{pp}, and two multivariate process centering indices, k_L and k_A. The basis of the proposal is that the actual and potential proportion nonconforming product can be measured, estimated, or calculated and then transformed into actual and potential capability indices. To ensure meaningful physical interpretations of the indices, the transformation was designed to ensure that the proposed index C_{p^*} reduces to the univariate PCI, C_p, when the underlying process follows a univariate normal distribution. Even though the proposed indices are stated as being general, they were designed around very specific process situations with elliptical or circular tolerances for hole location, coaxial hole pair, locations and angularity, and sets or systems of pairs of coaxial holes. The calculation of the proportion nonconforming is discussed under the multivariate normal assumption. It is stated that due to transformations of the variables, both coordinate system transformations and reference frame transformations, the multivariate assumption is not unreasonable and the data can be considered approximately normal. However, Rivera et al. (1995) concludes that estimation of proportion nonconforming after a transformation to normality is not a good approach, and leads to overestimates of the proportion nonconforming. Even though this would create a conservative estimate, the logarithmic transformation, for example, produced a 16 percent error, on average. The two multivariate process centering indices, k_L and k_A, were created to give an indication of the magnitude of any centering issues which may exist in the data. The subscript "L", denotes location variable centering, and the subscript "A", denotes angular orientation variable centering. Confidence statements on the proposed indices were created through bootstrap simulation.

Wierda (1993 & 1994) proposes the multivariate capability index, MC_{pk}, which is based on the multivariate normal distribution and rectangular specification limits. This multivariate extension of C_{pk} is calculated by an estimation of the percent of nonconforming product.

Chen (1994) proposes a measure of multivariate process capability, MC_p. It is stated that the index is the numerical ratio of the tolerance zone radius to the radius of the actual zone needed to achieve the desired expected proportion of nonconforming products. It is also stated that the proposed index is a natural generalization of the univariate PCI, C_p. I believe that the index is closer to the univariate PCI, C_{pk}, because the radii are truly functions of the target vector and the process data. The examples given, in fact, demonstrate that the index value decreases as the distance between the target and process mean increases. The original equations proposed can be applied to many situations, including both rectangular and elliptical specifications, as well as non-normal data, assuming that the distribution is known and certain properties can be estimated. Asymptotic confidence intervals are also discussed which are based on multivariate normal data and resampling estimates of the standard deviation of the approximated index value. Special cases of the PCI are discussed including one in which the index reduces to the multivariate PCI proposed by Pearn et al. (1992). Computation of the index can be very challenging, with the evaluation of a complex integral.

2.5 Summary

The review of existing literature indicates a need for a method which can evaluate the capability of a process consisting of variables which do not, necessarily, follow the normal distribution. The proposed indices which indicated non-normal applications, also require knowledge of the distribution and its parameters. Further, the distributions, although non-normal, would probably all have to be of the same type. In industry, processes which follow the normal distribution are not as common as statisticians would like. And, even though it is usually not very difficult to declare a process to be non-normal, it can be very difficult to identify the distributions which do represent the process. Similar types of dimensions from a multivariate process may follow similar distributions, however, many multivariate processes have very different types of dimensions, which would probably not follow the same distribution. This research presents a practical method to fill the identified gaps found in the current literature.

CHAPTER III

METHODOLOGY

3.1 Fitting a Marginal Distribution Sample with a Johnson System

3.1.1 Descriptive Statistics

There are a number of descriptive statistics which are used when fitting a marginal distribution sample with a Johnson system. The first four moments about the mean are calculated from the following equations:

$$m_1 = \bar{x} = \sum_{i=1}^{n} \frac{x_i}{n} \qquad m_2 = s^2 = \sum_{i=1}^{n} \frac{(x_i - m_1)^2}{n}$$

$$m_3 = \sum_{i=1}^{n} \frac{(x_i - m_1)^3}{n} \qquad m_4 = \sum_{i=1}^{n} \frac{(x_i - m_1)^4}{n}$$

where x_i represents the i^{th} measurement in a sample of n observations. The equation for m_1, as it is shown above, is actually the first moment about the origin, and not the first moment about the mean. The first moment about the mean is always equal to zero and thus gives no useful information. The first moment about the origin, m_1, is more commonly known as $\bar{x}$, or the sample average. The second moment about the mean, m_2, is more commonly known as s^2, or the sample variance, where s is the sample standard deviation. This equation should not be confused with the estimated population standard deviation which contains (n-1) in the denominator. Although it is difficult to give a physical meaning to the third and fourth moments about the mean, it is important to understand that they are used as descriptive statistics because they allow insight into the shape of the distribution.

Another common descriptive characteristic of a distribution is called skewness. Skewness represents the degree to which a distribution has non-symmetric tails. Positive skew depicts a distribution which has a "heavy" tail extending to the right. Conversely, negative skew depicts a distribution with a "heavy" tail extending to the left. There are two common dimensionless measures of skewness which are both functions of m_3 and can be calculated from the following equations:

$$\beta_1 = \frac{m_3^2}{m_2^3} \qquad \text{and} \qquad \gamma_1 = \sqrt{\beta_1} * \text{sign}(m_3)$$

Since β_1 is always positive, it does not reveal the direction of skewness. The direction of skewness, positive or negative, can be immediately determined by the sign of m_3, and the issue is resolved by giving this sign to the value of γ_1. Therefore, β_1, or its square root, can be interpreted as the magnitude of skewness which exists in the distribution, and the sign of γ_1 represents the direction to which the distribution is skewed.

Kurtosis is yet another common descriptive characteristic of a distribution. Kurtosis can be described as a measure of the flatness or peakedness of a distribution and is usually used when comparing symmetric distributions to a normal distribution. When a distribution is more flat than the normal distribution, it can be called platykurtic. Conversely, when a distribution is less flat, or more peaked, than the normal, it can be called leptokurtic. This definition has some problems because the skewness of a distribution also plays a large role in the value of kurtosis. Either way, there are two common dimensionless measures of kurtosis which are both functions of m_4 and can be calculated from the following equations:

$$\beta_2 = \frac{m_4}{m_2^2} \qquad \text{and} \qquad \gamma_2 = \beta_2 - 3$$

The value of β_2 for a normal distribution is exactly 3, and because it is often used in comparison, this value is subtracted in the equation for γ_2. Therefore, if γ_2 is negative, then the distribution is considered platykurtic, and if its value is positive, then the distribution is considered leptokurtic.

3.1.2 The Johnson System of Distributions

Before I can begin discussing the Johnson system of distributions, I must first cite two incredible references: Elderton and Johnson (1969) and Miller (1995a & 1995b). It is from these two references and personal mail with Dr. David Miller, that I am able to use the Johnson system for my research.

In 1949, Norman L. Johnson presented a set of three transformations used for fitting and / or explaining frequency curves. These transformations were meant to be alternatives to the set of 12 Pearson curves, introduced in 1893 by Karl Pearson. The Pearson curves can be difficult to use simply from their quantity and complexity. The skewness, β_1, and the kurtosis, β_2, are two key descriptive statistics which Johnson used to create his transformations. This pair of statistics, (β_1, β_2), is used to position any distribution on a two-dimensional plane. The three transformations were constructed to cover the entire (β_1, β_2) plane. This is to say that one of the transformations can assume any given pair of these values. Johnson presents his system with the statement that there is always a transformation function, f(x), which will produce a distribution which is exactly normal. However, due to the probable complexity of such a function, it would not be helpful. To ensure the usefulness of such a function, it would have to be relatively simple and consist of only a few parameters.

Johnson introduces two parameters, γ and δ, that are used to transform the normally distributed function, f(x), into a standardized normal variate, z.

$$z = \gamma + \delta f(x) \quad \sim \quad N(0,1) \quad , \delta > 0$$

This equation defines the values of β_1 and β_2 for the distribution. Two final parameters, ξ and λ, are introduced by replacing x with a non-constant linear function of x, $(x - \xi)/\lambda$. This gives us the general form of the Johnson transformation system:

$$z = \gamma + \delta f\left(\frac{x - \xi}{\lambda}\right) \quad \sim \quad N(0,1) \quad , \delta > 0$$

where values of the four parameters, γ, δ, ξ, and λ, determine the distribution of x.

3.1.2.1 Bounded Johnson System, S_B

The first transformation which Johnson presents is a system in which the variation of x is bounded. This system is called the S_B system, and is created by setting $f(x) = ln\{x / (1-x)\}$ where $(0 < x < 1)$. The four parameter system of curves is created by replacing x with $(x - \xi)/\lambda$:

$$S_B \text{ system: } z = \gamma + \delta\, ln\left[\frac{x - \xi}{\xi + \lambda - x}\right] \quad , (\xi < x < \xi + \lambda)$$

It can be seen from this equation that λ represents the range of variation of x, and that ξ and $(\xi + \lambda)$ represent the lower and upper bounds of the distribution, respectively. The density function for the S_B system is given here for reference only:

$$y_{S_B} = \frac{1}{\lambda\sqrt{2\pi}} \frac{1}{\left(\frac{x-\xi}{\lambda}\right)\left(1-\frac{x-\xi}{\lambda}\right)} e^{-\frac{1}{2}\left[\gamma+\delta \ln\left(\frac{x-\xi}{\xi+\lambda-x}\right)\right]^2} \quad , (\xi < x < \xi + \lambda)$$

Figure 3.1 shows a sample of the diverse shapes which the S_B system can take.

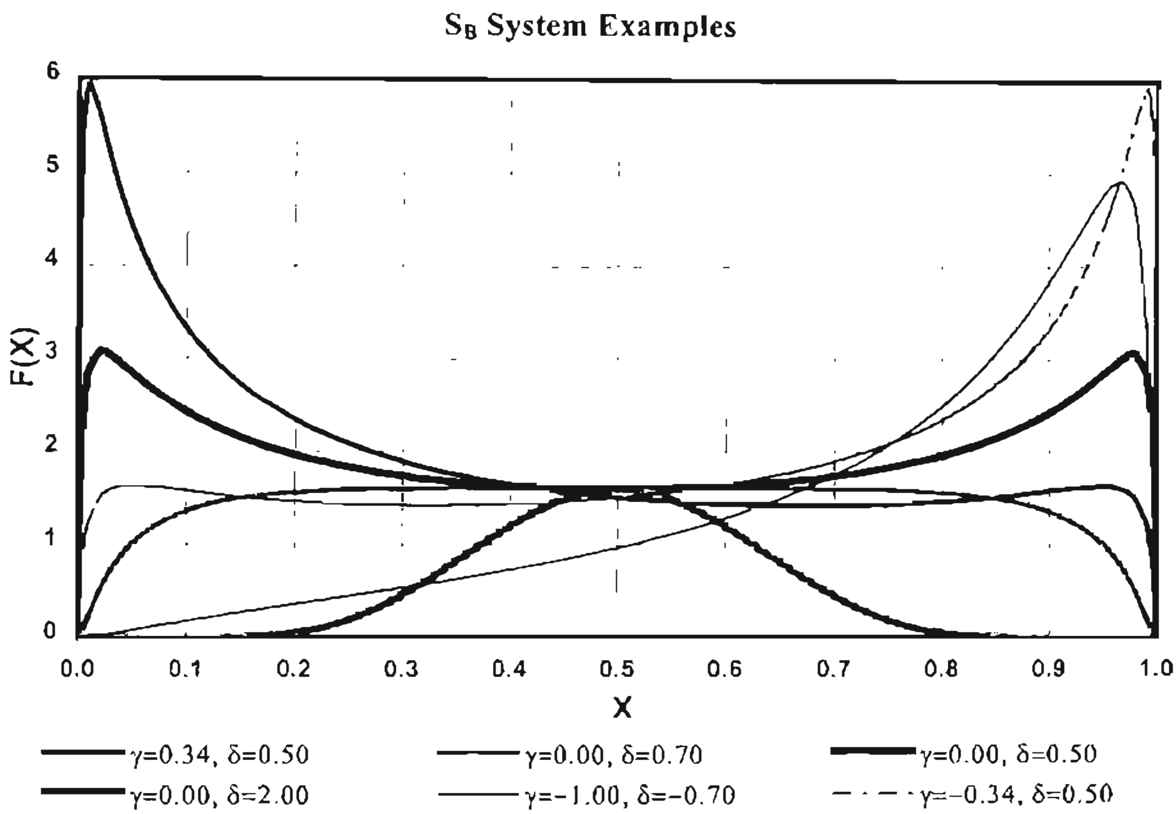


Figure 3.1: Density Plots for the S_B System (ξ=0, λ=1)

3.1.2.2 Unbounded Johnson System, S_U

The second transformation which Johnson presents is a system in which the variation of x is unbounded. This system is called the S_U system, and can be created by setting $f(x) = \sinh^{-1}(x) = ln\left(x + \sqrt{x^2 + 1}\right)$, which is the definition of the inverse hyperbolic sine. The four parameter system of curves is created by replacing x with $(x - \xi)/\lambda$:

$$S_U \text{ system:} \quad z = \gamma + \delta\, sinh^{-1}\left(\frac{x - \xi}{\lambda}\right) \qquad ,(-\infty < x < +\infty)$$

It can be seen from this equation that x is unbounded on both sides. The density function for the S_U system is given here for reference only:

$$y_{S_U} = \frac{1}{\lambda\sqrt{2\pi}} \frac{1}{\sqrt{1 + \left(\frac{x - \xi}{\lambda}\right)^2}} e^{-\frac{1}{2}\left[\gamma + \delta\, sinh^{-1}\left(\frac{x - \xi}{\lambda}\right)\right]^2} \qquad ,(-\infty < x < +\infty)$$

Figure 3.2 shows a sample of the shapes which the S_U system can take.

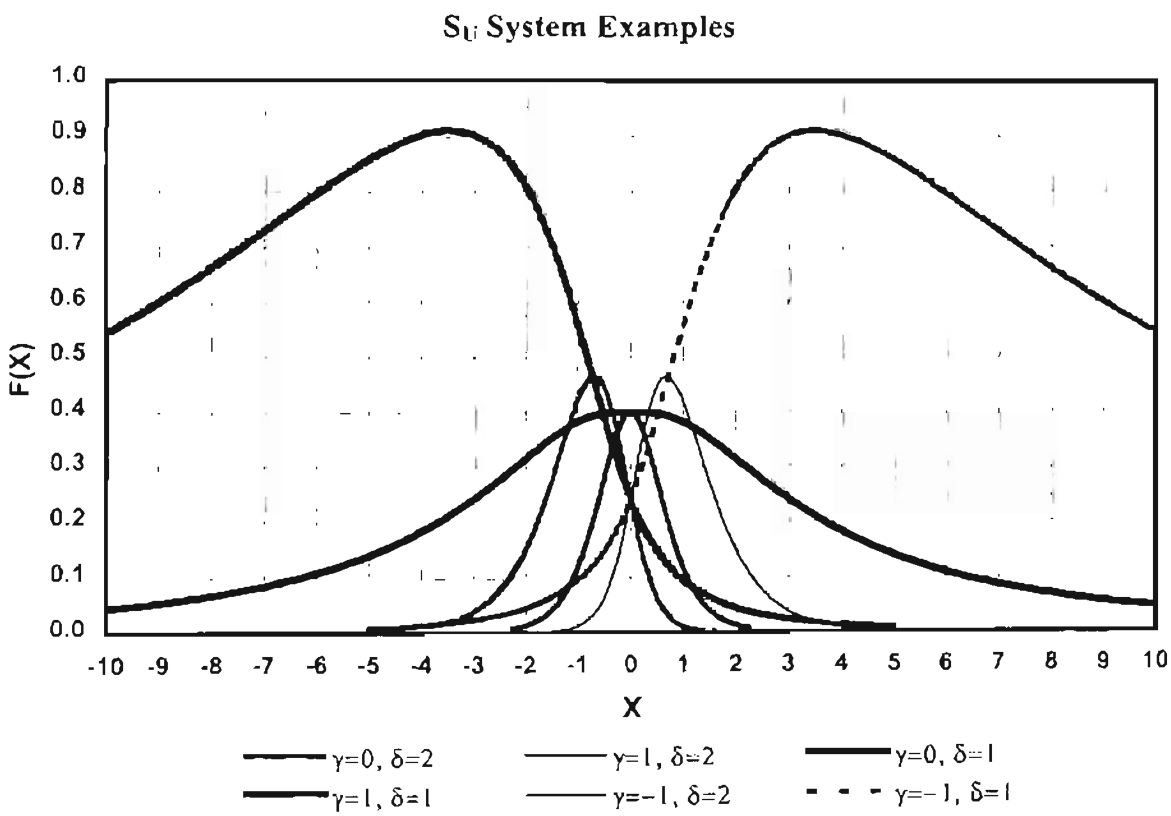


Figure 3.2: Density Plots for the S_U System (ξ=0, λ=1)

3.1.2.3 Lognormal Johnson System, S_L

Johnson presents a transition system between the S_B and S_U systems. It is stated that if $\delta \to \infty$ and γ is held constant, the S_B and S_U systems both tend to the normal as a limiting curve. However, if $\delta \to \infty$ and γ/δ is held constant, then the S_B and S_U systems both tend to the lognormal as a limiting curve. It is this lognormal system, called the S_L system, which is presented as the transition system between the S_B and S_U systems. This commonly-known system is created by setting $f(x) = ln(x)$. The four parameter system of curves, created by replacing x with $(x-\xi)/\lambda$, reduces to a three parameter system of curves:

$$S_L \text{ system: } z = \gamma + \delta\, ln(x-\xi) \qquad .(\xi < x < +\infty)$$

To avoid confusion, this system should probably be discussed further. First, note that the three parameter form of the lognormal is always positively skewed and bounded on the left by the location parameter, ξ. This can also reduce to the two parameter form of the lognormal when ξ is assumed to be zero, and is dropped from the equation. For obvious reasons, however, the three parameter form is used because of its greater generality. A simple way of explaining the three parameter form is that it occurs when the natural logarithm of $(x-\xi)$ is normally distributed. It was stated above that the four parameter system of curves reduces to a three parameter system of curves. This reduction can be seen if first, we were to write the four parameter form as follows:

$$z = \gamma' + \delta\, ln\left(\frac{x-\xi}{\lambda'}\right)$$

and then we reduce this to the three parameter form by allowing $\gamma = \gamma' - \delta\, ln(\lambda')$. The lambda parameter actually becomes part of the gamma parameter. Another way to keep the general four parameter form would be by holding $\lambda = 1$. The density function for the S_L system is given here for reference only:

$$y_{S_L} = \frac{1}{\sqrt{2\pi}} \frac{\delta}{(x-\xi)} e^{-\frac{1}{2}[\gamma - \delta \ln(x-\xi)]^2} \qquad . (\xi < x < +\infty)$$

Figure 3.3 shows a sample of the shapes which the S_L system can take.

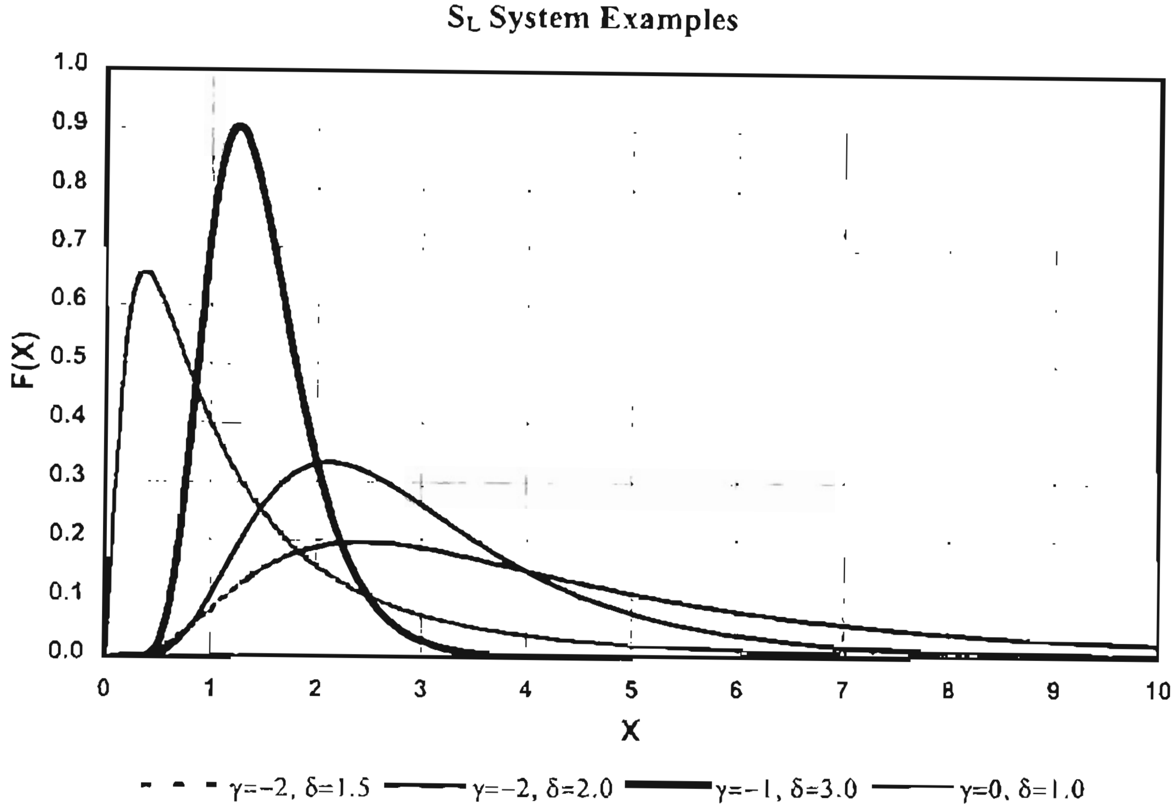


Figure 3.3: Density Plots for the S_L System (ξ=0)

3.1.2.4 Special Cases of the Johnson System, S_N and S_S

There are two special cases of the Johnson system which need to be discussed. The first special case is the normal distribution and will be referred to as the S_N system. This system occurs when x is normally distributed and can be created by setting $f(x) = x$. The four parameter system of curves, created by replacing x with $(x-\xi)/\lambda$, reduces to a two parameter system of curves:

$$S_N \text{ system: } z = \gamma + \delta x \qquad , (-\infty < x < +\infty)$$

It was stated above that the four parameter system of curves reduces to a two parameter system of curves. This reduction can be seen if first, we were to write the four parameter form as follows:

$$z = \gamma' + \delta'\left(\frac{x - \xi'}{\lambda'}\right)$$

and then we reduce this to the two parameter form by allowing $\delta = \delta'/\lambda'$ and $\gamma = \gamma' - \delta\xi'$. The remaining two parameters, γ and δ, are simple functions of the mean and the standard deviation of the distribution, where the mean equals $-\gamma/\delta$ and the standard deviation equals $1/\delta$. Another way to keep the general four parameter form would be by holding $\lambda = 1$ and $\xi = 0$. Note that it would have been just as easy, and tempting also, to allow the two parameters, ξ and λ, to correspond to the mean and the standard deviation of the distribution, while holding $\gamma = 0$ and $\delta = 1$. However, it was presented the other way to keep the same notation throughout the study. The density function for the S_N system is given here for reference only:

$$y_{S_N} = \frac{\delta}{\sqrt{2\pi}} e^{-\frac{1}{2}(\gamma + \delta x)^2} \qquad , (-\infty < x < +\infty)$$

Figure 3.4 shows a sample of the familiar shapes which the S_N system can take.

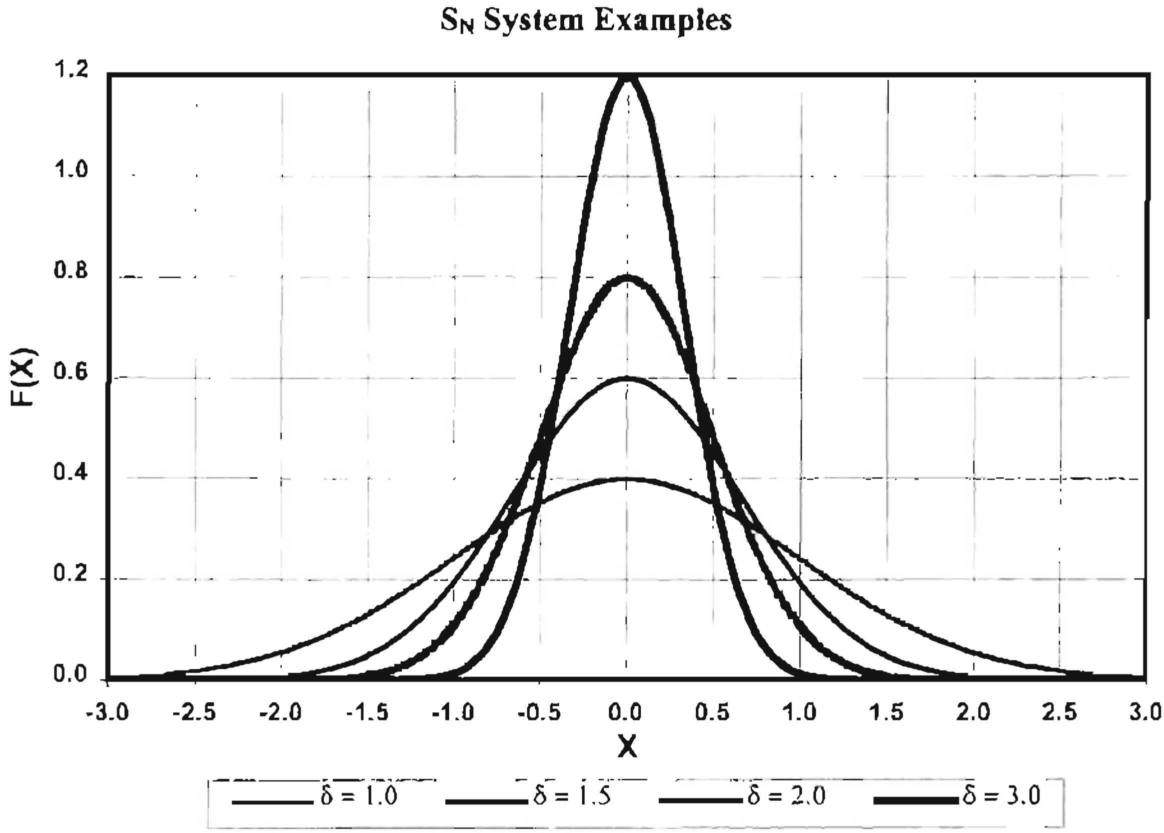


Figure 3.4: Density Plots for the S_N System (γ=0)

The other special case stems from a problem which I had with the S_L system. As was stated previously, the two and three parameter forms of the lognormal curve are strictly skewed to the right. To maintain generality, I created a new system which represents a negatively skewed lognormal curve, which I called the S_S system. The subscript, S, simply stands for "Special." This system is similarly created by setting f(x) = ln(x). The three parameter system of curves is created simply by replacing x with $(\xi - x)$ and is represented by the following equation:

$$S_S \text{ system: } z = \gamma + \delta\, ln(\xi - x) \quad , (-\infty < x < \xi)$$

Note that this special three parameter form of the lognormal is always negatively skewed and bounded on the right by the location parameter, ξ. This special form of the lognormal will not reduce to a two parameter form, unless the distribution was strictly negative. As with the S_L system, the general four parameter form can be kept by holding $\lambda = 1$. The density function for the S_S system is given here for reference only:

$$y_{S_S} = \frac{1}{\sqrt{2\pi}} \frac{\delta}{(\xi - x)} e^{-\frac{1}{2}[\gamma + \delta \ln(\xi - x)]^2} \qquad , (-\infty < x < \xi)$$

Figure 3.5 shows a sample of the shapes which the S_S system can take.

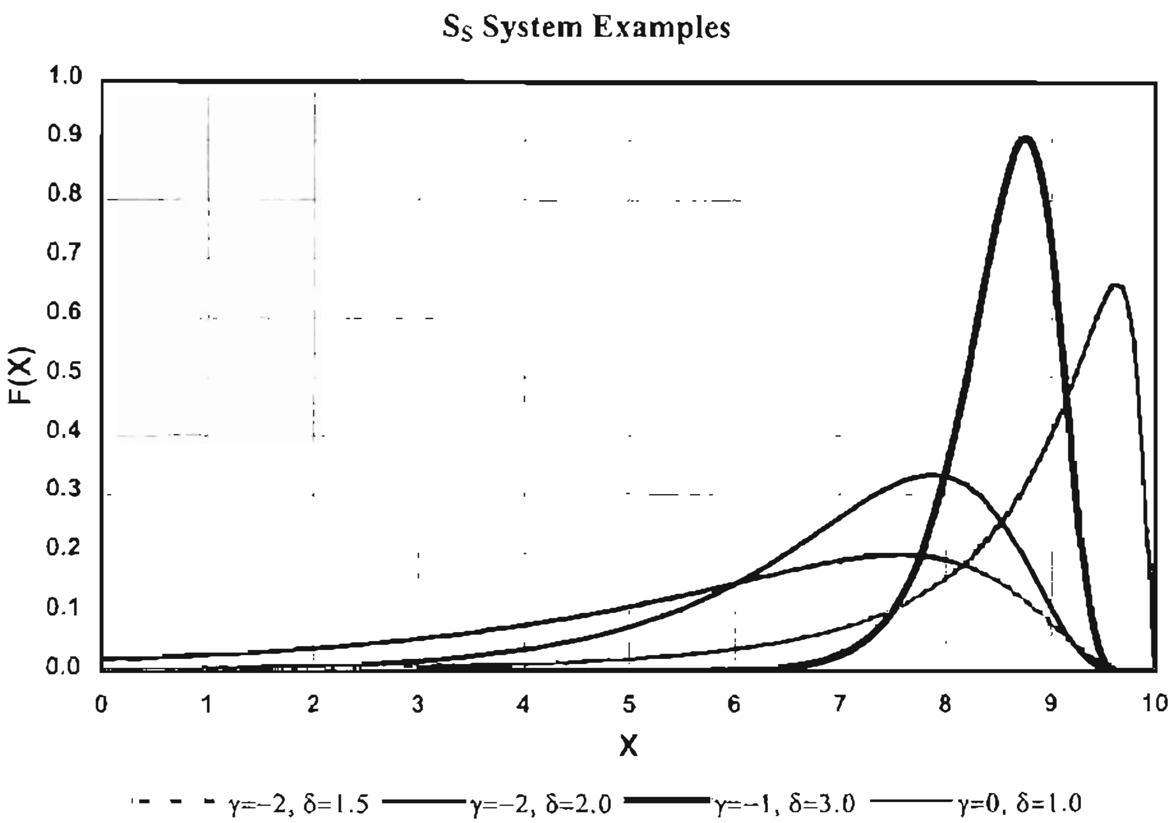


Figure 3.5: Density Plots for the S_S System (ξ=10)

3.1.2.5 The Lognormal Line

It was stated previously that a plane could be defined by the pair of statistics, (β_1, β_2), and that all distributions could be placed on this plane. The S_B and S_U systems were defined such that they would cover the entire (β_1, β_2) plane, with the S_L system acting as a transition between them. Johnson introduces a new parameter, ω, and defines two parametric equations for the lognormal distribution:

$$\beta_1 = (\omega - 1)(\omega + 2)^2 \quad \text{and} \quad \beta_2 = \omega^4 + 2\omega^3 + 3\omega^2 - 3$$

From these equations it can be seen that a (β_1, β_2) point is defined for a given value of ω. As ω is varied, a nearly straight line is produced across the (β_1, β_2) plane, which separates the plane into two distinct pieces. Johnson refers to this line as the lognormal line, and it separates S_B curves from S_U curves. The importance of this line is quite obvious when we desire to classify a given distribution. It should be noted that as the value of ω approaches 1, the normal curve is approached at point (0,3). Figure 3.6 graphically depicts the lognormal line on the (β_1, β_2) plane.

3.1.3 Distribution Classification

In the previous section, a set of parametric equations was presented which could be used to calculate values of β_1 and β_2, given a value of ω. The distinction between these two values and the values of the descriptive statistics, β_1 and β_2, is very important. When clarification is necessary, the word "calculated" will be used to identify those values of β_1 and β_2 calculated from the parametric equations of ω, and the word "observed" will be used to identify the descriptive statistics.

Miller proposes a method which determines the location of a given distribution, on the (β_1, β_2) plane, in relation to the lognormal line. The first step would be to solve for ω in the calculated β_1 equation by substituting in the observed β_1 value. Next, solve the calculated β_2 equation using this value of ω. If this calculated value of β_2 is less than the observed value of β_2, then the distribution resides on the S_U system side of the lognormal line. Conversely, if the calculated value of β_2 is greater than the observed value of β_2, then the distribution resides on the S_B system side of the lognormal line. The S_L system is used when both the calculated and observed values of β_2 are approximately equal. Miller presents a straightforward procedure for solving the cubic equation of β_1 for ω:

$$\omega = \frac{(C-1)^2}{C} + 1 \qquad \text{where} \qquad C = \left(\frac{2}{2 + \beta_1 + \sqrt{\beta_1 (4 + \beta_1)}} \right)^{1/3}$$

As an attempt to clear up any confusion which may exist, Figure 3.6 was created, which shows a portion of the (β_1, β_2) plane with the lognormal line and the S_B and S_U system areas identified.

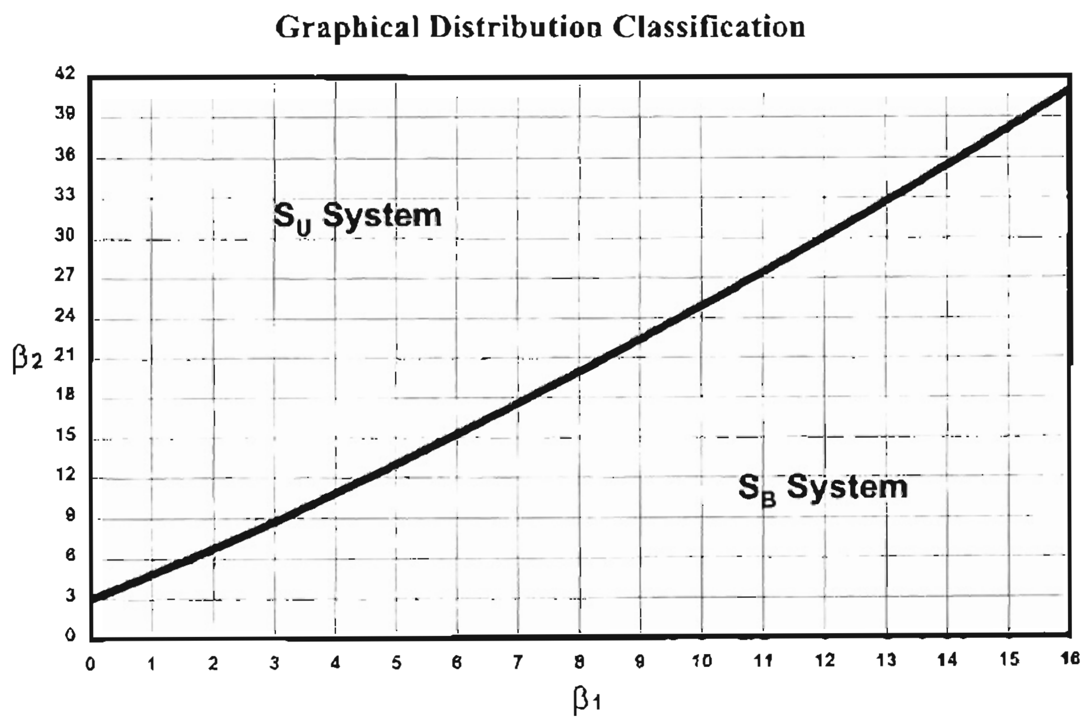


Figure 3.6: The Lognormal Line on the (β_1, β_2) Plane

Figure 3.6 could be used to approximate the distribution type by locating the point, (observed β_1, calculated β_2), in relation to the lognormal line.

3.1.4 Johnson System Parameter Estimation

3.1.4.1 S_N System Parameter Estimation

The S_N System requires two parameter estimates, γ and δ. The estimates are simple functions of the descriptive statistics and can be determined from the following equations:

$$\delta = \frac{1}{s} = \frac{1}{\sqrt{m_2}} \qquad \text{and} \qquad \gamma = -\frac{\bar{x}}{s} = -\frac{m_1}{\sqrt{m_2}}$$

As was stated previously, the general four parameter form of the S_N system can be used by setting $\lambda = 1$ and $\xi = 0$.

3.1.4.2 S_L System Parameter Estimation

The S_L system requires parameter estimates for γ, δ, and ξ. When the value of the location parameter, ξ, is unknown, numerous approaches for parameter estimation are available in current literature. A closed form estimate of the lognormal location parameter, ξ_E, is discussed on page 381 of Ebeling (1997). This estimate, using consistent notation, is as follows:

$$\xi_E = \frac{x_{min}\, x_{max} - x_{med}^2}{x_{min} + x_{max} - 2x_{med}} \qquad \text{where} \quad \begin{cases} x_{min} = \text{minimum sample value} \\ x_{max} = \text{maximum sample value} \\ x_{med} = \text{median sample value} \end{cases}$$

Miller defines another estimate derived from the equation for β_1 given in section 3.1.2.5. By taking the square root of both sides and utilizing the following substitutions,

$$t = \sqrt{\omega^2 - 1}, \quad \omega^2 = t^2 + 1, \quad \text{and} \quad \gamma_1 = \sqrt{\beta_1}$$

a cubic equation with only one real root is obtained. This equation is as follows:

$$t^3 + 3t - \gamma_1 = 0$$

A geometric solution for the single real root of this equation can be calculated from the following equation:

$$t = \sqrt[3]{\frac{\gamma_1}{2} + \sqrt{\frac{\gamma_1^2}{4} + \frac{(3)^3}{27}}} - \sqrt[3]{\frac{\gamma_1}{2} - \sqrt{\frac{\gamma_1^2}{4} + \frac{(3)^3}{27}}}$$

The estimate, ξ_M, of the lognormal location parameter, taken from Miller, is finally calculated from the following equation:

$$\xi_M = m_1 - \frac{\sqrt{m_2}}{t}$$

A third estimate of the lognormal location parameter, ξ_B, can be calculated from the following straightforward equation:

$$\xi_B = x_{min} - \frac{x_{max} - x_{min}}{n}$$

Through trial and error, a method for selecting which of the three location parameter estimates to be used was created. This method first decides whether the Ebeling estimate, ξ_E, and the Miller estimate, ξ_M, are realistic for the lognormal distribution. This "reality" check is done by determining whether the estimate is within an acceptable range to the left of the minimum sample value. If the check fails and finds that one or both of the estimates are unrealistic, then their values are set equal to the value of the ξ_B estimate. The minimum value of all realistic estimates is used as the initial estimate of the lognormal location parameter, ξ. This estimate can then be used in the Maximum Likelihood Estimation (MLE) equations for the remaining parameters, γ and δ. The following equations, using current notation, were found on page 122 of Johnson and Kotz (1970):

$$-\frac{\gamma}{\delta}=\frac{\sum_{i=1}^{n} ln(x_i-\xi)}{n} \quad \text{and} \quad \frac{1}{\delta}=\sqrt{\frac{\sum_{i=1}^{n}\left\{ln(x_i-\xi)-\left(-\frac{\gamma}{\delta}\right)\right\}^2}{n}}.$$

where the parameter estimates of γ and δ are found by simple manipulations of these equations. As was stated previously, the general four parameter form of the S_L system can be used by setting $\lambda = 1$. The two parameter form of the lognormal can be estimated by setting $\xi = 0$ in these equations. A quick check of whether the minimum sample value is to the right of zero, determines the feasibility of the two parameter form. This check prevents the two parameter form of the lognormal from being optimized if it is not feasible.

3.1.4.3 S_S System Parameter Estimation

The parameter estimates of γ, δ, and ξ for the S_S system can be determined by the S_L system equations with simple modifications. The modified Ebeling location parameter estimate, ξ_E, which now represents the right bound for the distribution, is required for the S_S system and can be calculated as follows:

$$\xi_E=\frac{|x_{min}\,x_{max}|+x_{med}^2}{|x_{min}|+|x_{max}|+2|x_{med}|} \qquad \text{where} \quad \begin{cases} x_{min} = \text{minimum sample value} \\ x_{max} = \text{maximum sample value} \\ x_{med} = \text{median sample value} \end{cases}$$

As with the S_L system, the Miller estimate of the location parameter, ξ_M, can be determined with the following equation modifications:

$$\xi_M = m_1 + \frac{\sqrt{m_2}}{|t_1|}$$

A third estimate of the lognormal location parameter, ξ_B, can be calculated from the following straight forward equation:

$$\xi_B = x_{max} + \frac{x_{max} - x_{min}}{n}$$

Location parameter estimate selection is carried out in the same manner as the S_L system method, with the exception that the checks are to the right of the maximum sample value instead of to the left of the minimum sample value. The maximum value of all realistic estimates is used as the initial estimate of the location parameter estimate, ξ. This estimate is then used in the MLE equations to determine the parameter estimates of γ and δ. These estimates are found by simple manipulations of the following equations:

$$-\frac{\gamma}{\delta} = \frac{\sum_{i=1}^{n} ln(\xi - x_i)}{n} \quad \text{and} \quad \frac{1}{\delta} = \sqrt{\frac{\sum_{i=1}^{n}\left\{ln(\xi - x_i) - \left(-\frac{\gamma}{\delta}\right)\right\}^2}{n}}$$

3.1.4.4 S_B System Parameter Estimation

Miller discusses the parameter estimation of the S_B system on pages 487-489. It is stated that numerical optimization is required to determine adequate estimates of the four parameters. Numerical optimization procedures usually require "ballpark" initial parameter estimates, and Miller presents a method for obtaining these estimates. Since ξ and $(\xi + \lambda)$ represent the lower and upper bounds of the distribution, respectively, it would seem logical to estimate the value of ξ as something slightly smaller than the minimum sample value and the value of ($\xi + \lambda$) as something slightly larger than the maximum sample value. The following two equations allow for such estimates:

$$\xi = x_{min} - \frac{x_{max} - x_{min}}{n} \qquad \text{and} \qquad \lambda = x_{max} + \frac{x_{max} - x_{min}}{n} - \xi$$

Once these estimates are known, the S_B distribution can be transformed into a normal distribution by substituting their values into the S_B function:

$$f_B = ln\left(\frac{x - \xi}{\xi + \lambda - x}\right)$$

The remaining two parameters, γ and δ, allow a final transformation into a standard normal distribution, utilizing the S_B system equation: $z = \gamma + \delta f_B$. If two points are selected from the data and their corresponding z values are calculated, two equations would be created that could be solved to determine the estimates of γ and δ. One such approach would be to select the points which represent the first and third quartiles of the data. The two equations would then be:

$$z_{0.25} = \gamma + \delta\, ln\left(\frac{x_{0.25} - \xi}{\xi + \lambda - x_{0.25}}\right) \qquad \text{where} \quad \begin{cases} z_{0.25} = 1^{st}\text{ quartile of the standard normal, } \Phi(z) \\ x_{0.25} = 1^{st}\text{ quartile of the sample} \end{cases}$$

$$z_{0.75} = \gamma + \delta\, ln\left(\frac{x_{0.75} - \xi}{\xi + \lambda - x_{0.75}}\right) \qquad \text{where} \quad \begin{cases} z_{0.75} = 3^{rd}\text{ quartile of the standard normal, } \Phi(z) \\ x_{0.75} = 3^{rd}\text{ quartile of the sample} \end{cases}$$

which can be solved simultaneously to determine the estimates of γ and δ. These four estimates of the parameters are sufficient to begin a numerical optimization procedure, which will be discussed later.

3.1.4.5 S_U System Parameter Estimation

Miller discusses the parameter estimation of the S_U system on pages 483-485. As with the S_B system parameter estimates, an approach is presented to calculate initial estimates of the S_U system parameters, which can then be used in a numerical optimization procedure. The procedure is iterative, and its steps, as presented, are as follows:

Step 1: Set $\delta = 4$, as an initial guess.

Step 2: Calculate $\omega = e^{(1/\delta^2)}$.

Step 3: For convenience, calculate $K = \dfrac{2(\beta_2 - 3)}{\omega - 1}$.

Step 4: Calculate three quantities which are functions of ω:

$$
\begin{aligned}
A_0 &= \omega^5 + 3\omega^4 + 6\omega^3 + 10\omega^2 + 9\omega + 3 \\
A_1 &= 8\left(\omega^4 + 3\omega^3 + 6\omega^2 + 7\omega + 3\right) \\
A_2 &= 8\left(\omega^3 + 3\omega^2 + 6\omega + 6\right)
\end{aligned}
$$

Step 5: Calculate:

$$m = \frac{\left[4K(\omega+1) - A_1\right] + \sqrt{\left[4K(\omega+1) - A_1\right]^2 - \left\{4\left(A_2 - 4K\right)\left[A_0 - K(\omega+1)^2\right]\right\}}}{2\left(A_2 - 4K\right)}$$

Step 6: Using these values of ω and m, calculate the value of β_1 that would be produced by ω and m. We will designate this values as $\tilde{\beta}_1$, where:

$$\tilde{\beta}_1 = \frac{m(\omega - 1)\left[4m(\omega + 2) + 3(\omega + 1)^2\right]^2}{2(2m + \omega + 1)^3}$$

The iterative procedure continues until this calculated value is essentially equal to the actual value. This iterative decision is accomplished by evaluating the difference between these two values:

$$\Delta = \left|\beta_1 - \tilde{\beta}_1\right| \qquad \text{where if} \quad \begin{cases} \Delta \le 1E-8 & \text{then skip to Step 9} \\ \Delta > 1E-8 & \text{then proceed to Step 7} \end{cases}$$

Step 7: For convenience, calculate:

$$C = 3 - 2\beta_2(1-\beta) - \beta(\omega^4 + 2\omega^2 + 3) \qquad \text{where} \quad \beta = \frac{\beta_1}{\tilde{\beta}_1}$$

Step 8: A new guess for ω is calculated from:

$$\omega = \sqrt{-1 + \sqrt{(1 - C)}}$$

With this new value of ω, return to Step 3.

Step 9: The parameter estimates are calculated from the following equations:

$$\delta = \sqrt{\frac{1}{ln(\omega)}} \qquad \gamma = \left|\delta\, sinh^{-1}\left(\sqrt{m}\right)\right| * \mathrm{sign}(m_3)$$

$$\lambda = \frac{\sqrt{m_2}}{\sqrt{\left(\frac{\omega - 1}{2}\right)\left[\omega\, cosh\left(\frac{2\gamma}{\delta}\right) + 1\right]}} \qquad \xi = m_1 + \lambda\sqrt{\omega}\, sinh\left(\frac{\gamma}{\delta}\right)$$

where *sinh* represents the hyperbolic sine function and *cosh* represents the hyperbolic cosine function. These four estimates of the parameters are sufficient to begin a numerical optimization procedure, which will be discussed later.

3.2 Goodness of Fit Test

A critical component of any curve fitting routine is the goodness of fit test. Stated simply, a goodness of fit test gives a numeric answer to the question of how well is the data explained by the specified distribution. The test results are not only used to determine how well the data fit the distribution, but also gives us a method of comparing the results from different distribution fits. Miller uses the chi-square, χ^2, test throughout his text as the goodness of fit test for frequency data. The χ^2 test can only be used on data which has been grouped into cells, while our data is one group of individual samples. If the sample size were sufficiently large, one could group the data into cells and then perform the χ^2 test. However, many samples taken in industry are small, and we must find a goodness of fit test that will apply to individual observations.

3.2.1 The Kolmogorov-Smirnov One-Sample Test

The Kolmogorov-Smirnov (K-S) one-sample test is a test of goodness of fit which applies to individual sample observations. In short, the K-S test compares the observed cumulative frequency distribution to the theoretical cumulative frequency distribution. The maximum deviation between the two distributions is calculated, and called the D-statistic. The sampling distribution of D is known and well documented, and tables of the critical values from that sampling distribution can be located in many references. The critical values of the D-statistic are a function of the sample size. This is due to the fact that sampling error should decrease as the sample size increases, which means that the maximum allowable deviation between the observed and theoretical distributions decreases as the sample size increases. A useful reference on the Kolmogorov-Smirnov one-sample test is Siegel (1956).

Pomeranz (1974) presents a function called PKS2 which calculates the exact probability of obtaining a deviation less than D. This value is often referred to as the p-value of the D-statistic, and can be used as the numerical goodness of fit value. As the D-statistic approaches zero, the observed distribution approaches the theoretical distribution and the p-value approaches zero. I personally like to transform the p-value into an f-value by the transformation: f-value = 1 – p-value. The f-value allows me to use the "bigger is better" viewpoint. That is to say that the f-value approaches one as the observed distribution approaches the theoretical distribution. If, for example, we were to specify a 90% significance level, then we would reject any fit with a f-value less than 0.90 or a p-value greater than 0.10.

3.2.1.1 K-S Test Application on the Johnson Systems

The Johnson system equations have all been presented in the form of the variable z, which if the data were adequately fit by the Johnson system, would be distributed as the standard normal, with a mean of zero and a standard deviation of one. Therefore, we will specify the theoretical frequency distribution as the standard normal distribution, and the distribution of variable z, as the observed frequency distribution. The observed cumulative frequency distribution can be determined from the following equation:

$$O_{CFD}(z) = \frac{k}{n} \qquad \text{where} \quad k = \text{ the number of observations} \leq z$$

Since the theoretical frequency distribution is the standard normal distribution, the theoretical cumulative frequency distribution is standard reference in most statistical texts and can be determined from the following equation:

$$T_{CFD}(z) = \Phi(z) \qquad \text{where} \quad \Phi(z) = \int_{-\infty}^{z} \left(1/\sqrt{2\pi}\right) e^{-y^2/2}\, dy$$

The D-statistic is calculated from the following equation:

$$D = \text{maximum}\left|T_{CFD}(z_i) - O_{CFD}(z_i)\right| \qquad \text{for } i = 1,\ldots,n$$

The values of D and n can be input into the PKS2 function, previously discussed. The p-value output can then be used to determine the goodness of fit between the sample data and the Johnson system.

3.3 Numerical Parameter Optimization

It was stated previously that the parameter estimates for the S_B and S_U systems were initial estimates, and required numerical optimization. Most numerical optimization procedures attempt to maximize or minimize a given function with any given variables and constraints. The Johnson transformation, in its general form, contains four variables and some constraints. The function which is minimize or maximize, however, needs to be discussed further. Most references which discuss parameter estimation use MLE equations, which are stated as being the absolute best estimates. MLE parameter estimates are developed by maximizing the likelihood function, which in simple terms can be stated as maximizing the probability of observing the data exactly as it was observed. Miller states that the function which is optimized should maximize the measure of goodness of fit. Miller uses the χ^2 test as his measure of goodness of fit, and presents a second function option. Since a smaller value of χ^2 represents a "better" fit, he introduces the Minimum Chi-square Estimation (MCE) as an option to MLE. It makes sense that when the value of χ^2 is minimized, then the parameters will be optimized to give a maximum measure of goodness of fit. It is stated that MLE estimates are still recognized as being the best, however, a direct relationship exists between MCE and MLE estimates, which allow either to be used. I believe the reason that MCE was even considered was due to the complexity of the MLE equations required in the Johnson system functions. As was discussed previously, Chi-square, and thus MCE, will not work for data which are not grouped into cells. The K-S test's D-statistic is used in this research as a measure of goodness of fit because of its ability to handle individual measurements. As with the χ^2 value, a smaller value of the D-statistic represents a "better" fit, and therefore, using similar logic as Miller, we can present the Minimum D-statistic Estimation (MDE) as an alternative to the complex MLE equations.

Nelder and Mead (1965) present a simplex method for function minimization, and an algorithm based on this method is presented by O'Neill (1971). The algorithm allows a given continuous multivariate function to be minimized over all of the variables. When all variables are unconstrained, multivariate function minimization works very well. However, when parameter constraints exist, represented as bounds with the S_B, S_U, S_L, and S_S systems, a technique for maintaining the "continuous" requirement of the function must be developed. Alternatively, the initial estimates of the constrained parameters can be held constant, while the remaining parameters are allowed to change with the optimization. This second method does not allow the MDE equations to be completely optimized, however, Miller presents this as an option due to the significant increase in computation time when all variables are optimized, and because the constrained parameters of the Johnson systems are less critical to the goodness of fit.

3.3.1 Johnson System Parameter Optimization

3.3.1.1 S_N System Parameter Optimization

Both parameter estimates of δ and γ of the S_N system are essentially MLE estimates, and thus do not require optimization.

3.3.1.2 S_L System Parameter Optimization

The parameter estimates of ξ, δ, and γ for the S_L system are optimized using both MDE and a method of minimizing the negative natural logarithm of the S_L density function. This second optimization "function" is a variant of maximizing the log likelihood, and the method will be referred to as MLL. Both MDE and MLL optimization functions are used independently, primarily due to their differing performances in certain situations. If the two parameter form of the S_L system was found to be feasible in the parameter estimation phase, then it can also be optimized with MDE and MLL functions by setting $\xi = 0$ and holding its value constant. After parameter optimization, we have three sets of parameters, and if the two parameter form was used, then six sets of parameters exist for the S_L system. Selection of the "best" S_L system parameter set is determined by the set which yields the maximum K-S test f-value.

3.3.1.3 S_S System Parameter Optimization

As was stated previously, the S_S system represents a special case of the S_L system, and thus its parameters are optimized very similarly. The parameter estimates of ξ, δ, and γ for the S_S system are optimized using both MDE and MLL methods. Unlike the S_L system, the two parameter form of the S_S system is not considered. Therefore, three sets of parameters exist for the S_S system after optimization, and like the S_L system, selection of the "best" parameter set is determined by the set which yields the maximum K-S test f-value.

3.3.1.4 S_B System Parameter Optimization

The parameter estimates of δ and γ of the S_B system are optimized using MDE. The initial estimate of ξ, which is the lower bound of the system, and the initial estimate of λ, which is the distance between the lower and upper bounds of the system, are both held constant during the optimization procedure.

3.3.1.5 S_U System Parameter Optimization

The parameter estimates of δ, γ, ξ, and λ of the S_U system are all optimized using MDE. This can be done easily because the S_U system is completely unbounded, and therefore its parameters are unconstrained.

3.4 Johnson System Selection

There are five Johnson systems which have been discussed in this study: S_N, S_B, S_U, S_L, and S_S. However, all five cannot be fitted to a marginal distribution sample, because some of the systems compete with each other. The S_L and S_S systems can be considered competing with each other because the S_L system can only be fit to positively skewed data and the S_S system can only be fit to negatively skewed data. The S_B and S_U systems can be considered competing with each other because they reside on opposite sides of the lognormal line in the (β_1, β_2) plane. It would seem possible, due to sampling error, to select the wrong system when the distribution lies close to the lognormal line. However, the S_L or S_S system fit, being defined as transitional, should allow for this situation. The S_N system doesn't compete with another, and therefore can be used to fit any sample data.

Any given marginal distribution sample, therefore, can only be fit with three of the five Johnson systems, based completely on the sample's descriptive statistics. Further, due to the system competition, there are only four sets of three systems possible. These are the (S_N, S_L, S_B) set, the (S_N, S_L, S_U) set, the (S_N, S_S, S_B) set, and the (S_N, S_S, S_U) set. Due to the wide range of shapes which the S_B and S_U systems can take, they become very powerful at fitting sample data. Believe it or not, this causes some significant problems in the system selection process, particularly when the S_B system is available. Whenever a system with bounds "better fits" sample data from a distribution without bounds, the calculated capability performance of the distribution is greatly inaccurate. This is primarily seen when the specification limits lie outside of the bounded distribution's location parameters, and the proportion of nonconforming product is zero on that particular side of the distribution. The S_L and S_S systems exhibit this same problem, to some degree, when they "better fit" sample data from a normal distribution. Extensive study was used to develop a decision method which minimizes the risk of incorrectly identifying sample data from a normal distribution.

3.4.1 Test for Normality

Miller presents a very simple test for normality based on the measure of skewness. Since the normal distribution is not skewed, the theoretical value of γ_1 is zero. A hypothesis test is set up which tests whether it is possible that the observed value of γ_1 could have came from a normal distribution based on sampling error. The standard error of γ_1 for random samples drawn from a normal population is needed, and Miller presents the following equation for its calculation:

$$\sigma_{\gamma_1} = \sqrt{\frac{6}{n}}$$

A z-value and a two-tailed probability can be calculated which can be used to reject or fail to reject the null hypothesis, which states that $\gamma_1 = 0$, or that the sample came from a normal process. The z-value, Z_γ, can be calculated from the following equation:

$$Z_\gamma = \frac{|\gamma_1|}{\sigma_{\gamma_1}} \qquad \text{where} \qquad Z_{crit} = 1.96\ (95\%)$$

Therefore, when $Z_\gamma < 1.96$, we cannot reject the null with 95% confidence.

3.4.2 System Selection Decision Method

The decision method starts by calculating the f-value for the S_N system fit, F_N, and the f-value for the optimal S_L or S_S system's fit, F_{LS}. These f-values are considered "low" if they are less than 0.2, and "high" if they are greater than 0.2. A new variable, Δ, is introduced which minimizes some of the error of selecting the S_L or S_S system fit when the sample comes from a normal population. This variable is defined as follows:

$$\Delta = F_{LS} - F_N$$

The value of Δ is considered "high" if it is greater than 0.3, and "low" if it is less than 0.3. The last decision variable which is used is the z-value for γ_1, Z_γ. The value of Z_γ is considered "low" if it is less than 1.96, and "high" if it is greater than 1.96. There are normally 16 possible combinations of four variables, however, due to the definition of Δ, only 12 combinations are possible. These 12 combinations and their respective system selection decision are listed in Table 3.1. Please note that when two system selections are listed, the systems are competing, based on skew or the lognormal line, and the decision is based on the sample descriptive statistics. If the decision variables point towards the S_B or S_U system, then the procedure goes on to estimate and optimize the parameters for that particular system. Note that the "critical" values of the decision variables were based on trial and error, and their selected values were chosen to ensure a high probability of selecting the S_N system for a sample which came from a normal population. This selection matrix is somewhat sensitive to the "critical" values of the decision variables. Further study is warranted in this area.

System Selection Decision Table

$F_{N_{LOW}}$	$F_{LS_{LOW}}$	Δ_{LOW}	$Z_{Y_{LOW}}$	$\Rightarrow$	S_B, S_U
$F_{N_{LOW}}$	$F_{LS_{LOW}}$	Δ_{LOW}	$Z_{Y_{HIGH}}$	$\Rightarrow$	S_B, S_U
$F_{N_{LOW}}$	$F_{LS_{HIGH}}$	Δ_{LOW}	$Z_{Y_{LOW}}$	$\Rightarrow$	S_B, S_U
$F_{N_{LOW}}$	$F_{LS_{HIGH}}$	Δ_{LOW}	$Z_{Y_{HIGH}}$	$\Rightarrow$	S_L, S_S
$F_{N_{LOW}}$	$F_{LS_{HIGH}}$	Δ_{HIGH}	$Z_{Y_{LOW}}$	$\Rightarrow$	S_B, S_U
$F_{N_{LOW}}$	$F_{LS_{HIGH}}$	Δ_{HIGH}	$Z_{Y_{HIGH}}$	$\Rightarrow$	S_L, S_S
$F_{N_{HIGH}}$	$F_{LS_{LOW}}$	Δ_{LOW}	$Z_{Y_{LOW}}$	$\Rightarrow$	S_N
$F_{N_{HIGH}}$	$F_{LS_{LOW}}$	Δ_{LOW}	$Z_{Y_{HIGH}}$	$\Rightarrow$	S_B, S_U
$F_{N_{HIGH}}$	$F_{LS_{HIGH}}$	Δ_{LOW}	$Z_{Y_{LOW}}$	$\Rightarrow$	S_N
$F_{N_{HIGH}}$	$F_{LS_{HIGH}}$	Δ_{LOW}	$Z_{Y_{HIGH}}$	$\Rightarrow$	S_L, S_S
$F_{N_{HIGH}}$	$F_{LS_{HIGH}}$	Δ_{HIGH}	$Z_{Y_{LOW}}$	$\Rightarrow$	S_B, S_U
$F_{N_{HIGH}}$	$F_{LS_{HIGH}}$	Δ_{HIGH}	$Z_{Y_{HIGH}}$	$\Rightarrow$	S_L, S_S

Table 3.1: Decision Table for Johnson System Selection

This system selection decision method for fitting a marginal distribution sample is completed independently for each marginal distribution sample contained in the study. When the selection procedure is complete, each marginal distribution will be identified by an individual Johnson system and its required estimated parameters.

3.5 Multivariate Johnson Simulation

With each of the marginal distributions identified by a Johnson system, we have essentially identified a multivariate Johnson distribution. Further, when the marginal distributions have been transformed with their respective Johnson equations, we are left with a multivariate distribution with all of its variables distributed as the standard normal. Elderton and Johnson suggest, on pages 147 – 148, that it is a reasonable assumption that the joint distribution of standard normal marginal distributions will be multivariate standard normal. This is certainly not a guarantee, however, without any contrary knowledge, it is a reasonable approximation.

At this point, with a multivariate normally distributed process, it would appear that we are in a position to allow existing literature to define the multivariate process capability index. Unfortunately, there are a number of problems that would be encountered if that approach were taken. When data are transformed into the normal distribution, many times the shape of the distribution will significantly change. It seems logical that if the original distribution were either heavy-tailed or bounded, then as the distribution were transformed to the normal, the tail probabilities would change. This would ultimately lead to an inaccurate estimate of the proportion nonconforming. In fact, Rivera et al. (1995) demonstrate that the lognormal transformation yields a fairly consistent 16% error in the estimate of proportion nonconforming. They conclude that calculating a point estimate of the proportion nonconforming from a transformed distribution is not a good idea. It should be noted at this point, that some of the S_B and S_U distribution shapes are much less normal-like than the S_L or S_S distributions. This would suggest that larger errors could occur with other Johnson transformations. These errors of estimation are not acceptable, and therefore, another approach must be considered.

We are left with the requirement of estimating the proportion nonconforming from the original observed distributions. Unlike the normal distribution however, the standardized Johnson distribution does not exist, and therefore, the cumulative probabilities would have to be determined for each and every observed distribution. As a solution to this problem, we can simulate a multivariate standard normal distribution and reverse the transformations to allow us to reproduce the original multivariate Johnson distribution. The simulation method allows the proportion nonconforming to be calculated, as well as appropriate confidence intervals. The method is very straight forward and only requires knowledge of the transformed distribution correlation, the conditional multivariate Johnson equations, and the reverse Johnson transformations. Multivariate Johnson distribution simulation is discussed in chapter five of Johnson (1987).

3.5.1 Distribution Correlation

The correlation, ρ_{ij}, between each pair of marginal distributions is simply a measure of the dependence between the two distributions. Knowledge of the correlations is very important, and it is probably the sole reason why multivariate process capability indices are needed. This is to say that univariate process capability indices can be very misleading when used on variables which are dependent and correlated.

Simulation of a multivariate Johnson distribution requires knowledge of the correlation between each transformed marginal distribution. It is important to recognize that the correlation between the transformed variables, z_i and z_j, will be different from the correlation between the observed variables, x_i and x_j. However, the correlations values are related through the transformation. Before we can calculate the correlation, ρ_{ij}, we must first calculate the sample covariance, S_{ij}, and the sample variances, S_{ii} and S_{jj}. The sample covariance is calculated from the following equation:

$$S_{ij} = S_{ji} = \mathrm{Cov}(z_i, z_j) = \sum_{k=1}^{n} \frac{(z_{ik} - \bar{z}_i)(z_{jk} - \bar{z}_j)}{n} \qquad \text{where} \quad i \neq j$$

and the sample variances are calculated from the following equations:

$$S_{ii} = s_i^2 = \mathrm{Var}(z_i) = \sum_{k=1}^{n} \frac{(z_{ik} - \bar{z}_i)^2}{n} \qquad \text{and} \qquad S_{jj} = s_j^2 = \mathrm{Var}(z_j) = \sum_{k=1}^{n} \frac{(z_{jk} - \bar{z}_j)^2}{n}$$

Note that they are calculated the same way as the second moment about the mean, m_2. The only difference, of course, is that m_2 is the sample variance of the observed variable x, and not of the transformed variable, z. The variances and covariances are often described in matrix notation, where the matrix is simply called the variance-covariance matrix, Σ. The variance-covariance matrices for two, three, and four variables are as follows:

$$\Sigma_2 = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \qquad \Sigma_3 = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \qquad \Sigma_4 = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & S_{22} & S_{23} & S_{24} \\ S_{31} & S_{32} & S_{33} & S_{34} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{bmatrix}$$

Note that the variance-covariance matrices are diagonal and symmetric. The correlation can be calculated from the following equation:

$$\rho_{ij} = \rho_{ji} = \mathrm{Corr}(z_i, z_j) = \frac{S_{ij}}{\sqrt{S_{ii}}\sqrt{S_{jj}}} = \frac{S_{ij}}{s_i\, s_j} \qquad ,(-1 \leq \rho_{ij} \leq 1)$$

Note that when $i = j$, then $\rho_{ii} = \rho_{jj} = 1$. This makes sense when we consider that any given variable would be perfectly correlated with itself. The correlations are also often described in matrix notation, where the matrix is simply called the correlation matrix, $\mathbf{R}$. The correlation matrices for two, three, and four variables are as follows:

$$\mathbf{R}_2 = \begin{bmatrix} 1 & \rho_{12} \\ \rho_{21} & 1 \end{bmatrix} \qquad \mathbf{R}_3 = \begin{bmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{21} & 1 & \rho_{23} \\ \rho_{31} & \rho_{32} & 1 \end{bmatrix} \qquad \mathbf{R}_4 = \begin{bmatrix} 1 & \rho_{12} & \rho_{13} & \rho_{14} \\ \rho_{21} & 1 & \rho_{23} & \rho_{24} \\ \rho_{31} & \rho_{32} & 1 & \rho_{34} \\ \rho_{41} & \rho_{42} & \rho_{43} & 1 \end{bmatrix}$$

A convenient relationship can be obtained between the correlation matrix, $\mathbf{R}$, and the variance-covariance matrix, Σ, if we first define the standard deviation matrix, $\mathbf{V}^{1/2}$. The standard deviation matrices for two, three, and four variables are as follows:

$$\mathbf{V}_2^{1/2} = \begin{bmatrix} \sqrt{S_{11}} & 0 \\ 0 & \sqrt{S_{22}} \end{bmatrix} \qquad \mathbf{V}_3^{1/2} = \begin{bmatrix} \sqrt{S_{11}} & 0 & 0 \\ 0 & \sqrt{S_{22}} & 0 \\ 0 & 0 & \sqrt{S_{33}} \end{bmatrix}$$

$$\mathbf{V}_4^{1/2} = \begin{bmatrix} \sqrt{S_{11}} & 0 & 0 & 0 \\ 0 & \sqrt{S_{22}} & 0 & 0 \\ 0 & 0 & \sqrt{S_{33}} & 0 \\ 0 & 0 & 0 & \sqrt{S_{44}} \end{bmatrix}$$

The relationships can then be defined as follows:

$$\Sigma = \mathbf{V}^{1/2}\,\mathbf{R}\,\mathbf{V}^{1/2} \qquad \text{and} \qquad \mathbf{R} = \left(\mathbf{V}^{1/2}\right)^{-1} \Sigma \left(\mathbf{V}^{1/2}\right)^{-1}$$

A very interesting simplification occurs when the marginal distributions are standard normally distributed:

$$\Sigma = \mathbf{R} \qquad \text{when } S_{ii} = 1 \qquad \text{for } i = 1,\ldots,p$$

where p is the number of variables.

3.5.2 Conditional Multivariate Johnson Distributions

A complete discussion of the bivariate Johnson distribution, including the conditional equation development, can be found on pages 147 – 150 of Elderton and Johnson. Since this research is limited to four variables, we concern ourselves with four marginal distributions which have each been fit by a Johnson system. In order to present in generality, we represent the first marginal distribution, $f(x_1)$, by S_I, the second, $f(x_2)$, by S_J, the third, $f(x_3)$, by S_K, and the fourth, $f(x_4)$, by S_M. The subscripts I, J, K, and M can each be N, L, B, or U, representing the assigned Johnson system. Note that S_S is a special case of S_L and therefore, need not be identified separately, at this point. For convenience we denote $(x-\xi)/\lambda$ by x'. The four Johnson functions can then be written as follows:

$$
\begin{array}{lcl}
f_N(x') = x' & & z = \gamma + \delta f_N(x') \\
f_L(x') = ln(x') & & z = \gamma + \delta f_L(x') \\
f_B(x') = ln\left(\dfrac{x'}{1-x'}\right) & \text{where} & z = \gamma + \delta f_B(x') \\
f_U(x') = sinh^{-1}(x') & & z = \gamma + \delta f_U(x')
\end{array}
$$

Using the well known properties of the standardized multivariate normal distribution, the expected value of the conditional multivariate normal distribution can be calculated. Three good references for the multivariate normal distribution are Johnson and Wichern (1992), Tong (1990), and Johnson and Kotz (1972). If we let:

$$
\mathbf{Z} = \left[\begin{array}{c} \mathbf{Z}_1 \\ \hdashline \mathbf{Z}_2 \end{array}\right], \qquad \Sigma = \left[\begin{array}{c:c} \Sigma_{11} & \Sigma_{12} \\ \hdashline \Sigma_{21} & \Sigma_{22} \end{array}\right] \quad \text{and} \quad \mathbf{R} = \left[\begin{array}{c:c} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \hdashline \mathbf{R}_{21} & \mathbf{R}_{22} \end{array}\right]
$$

then the conditional distribution of $\mathbf{Z}_1$, given $\mathbf{Z}_2 = z_2$, is normal with mean = $\Sigma_{12}\,\Sigma_{22}^{-1}\,z_2$ and standard deviation = $\left[\Sigma_{11} - \Sigma_{12}\,\Sigma_{22}^{-1}\,\Sigma_{21}\right]^{1/2}$. This is to say that the expected value of the conditional multivariate standard normal is as follows:

$$\mathbf{E}\left[\mathbf{Z}_1 \mid \mathbf{Z}_2\right] = \Sigma_{12}\,\Sigma_{22}^{-1}\,\mathbf{Z}_2$$

Recalling that $\Sigma = \mathbf{R}$ for the multivariate standard normal distribution, we can write the expected value in terms of the correlation:

$$\mathbf{E}\left[\mathbf{Z}_1 \mid \mathbf{Z}_2\right] = \mathbf{R}_{12}\,\mathbf{R}_{22}^{-1}\,\mathbf{Z}_2$$

The variance of the conditional multivariate standard normal is as follows:

$$\mathbf{Var}\left[\mathbf{Z}_1 \mid \mathbf{Z}_2\right] = \Sigma_{11} - \Sigma_{12}\,\Sigma_{22}^{-1}\,\Sigma_{21}$$

and in terms of the correlation:

$$\mathbf{Var}\left[\mathbf{Z}_1 \mid \mathbf{Z}_2\right] = \mathbf{R}_{11} - \mathbf{R}_{12}\,\mathbf{R}_{22}^{-1}\,\mathbf{R}_{21}$$

3.5.2.1 Conditional Bivariate Johnson Equations

We define a bivariate Johnson distribution of type S_{IJ} by requiring the joint distribution of $z_1 = \gamma_1 + \delta_1\, f_I(x_1')$ and $z_2 = \gamma_2 + \delta_2\, f_J(x_2')$ to be described by a standardized bivariate normal distribution. There are 16 different S_{IJ} distributions based on the four Johnson systems. They are S_{NN}, S_{LN}, S_{BN}, S_{UN}, S_{NL}, S_{LL}, S_{BL}, S_{UL}, S_{NB}, S_{LB}, S_{BB}, S_{UB}, S_{NU}, S_{LU}, S_{BU}, and S_{UU}. Of these, S_{NN} represents an observed bivariate normal distribution. Using the well known properties of the standardized bivariate normal distribution, the conditional bivariate Johnson equations can be developed. The conditional distribution of $z_2 = \gamma_2 + \delta_2\, f_J(x_2')$, given z_1, is also normally distributed with an expected value of $\rho_{12} z_1$ and a standard deviation of $\sqrt{1-\rho_{12}^2}$. If we rearrange and partition the variable array and the correlation matrix as follows:

$$\mathbf{Z} = \left[\begin{array}{c} z_2 \\ \hline z_1 \end{array}\right] \qquad \text{and} \qquad \mathbf{R} = \left[\begin{array}{c|c} 1 & \rho_{21} \\ \hline \rho_{12} & 1 \end{array}\right]$$

and use the expected value equation, $\mathbf{E}[\mathbf{Z}_1 \mid \mathbf{Z}_2] = \mathbf{R}_{12}\,\mathbf{R}_{22}^{-1}\,\mathbf{Z}_2$, then the expected value of z_2, given z_1, can be calculated as follows:

$$\mathbf{E}(z_2 \mid z_1) = \rho_{21}\,(1)^{-1}\,z_1 \quad = \quad \rho_{12}\,z_1 \qquad \text{where} \quad \rho_{12} = \rho_{21}$$

If we apply the variance equation, $\mathbf{Var}[\mathbf{Z}_1 \mid \mathbf{Z}_2] = \mathbf{R}_{11} - \mathbf{R}_{12}\,\mathbf{R}_{22}^{-1}\,\mathbf{R}_{21}$, then the variance of z_2, given z_1, can be calculated as follows:

$$\mathbf{Var}(z_2 \mid z_1) = 1 - \rho_{21}\,(1)^{-1}\,\rho_{12} \quad = \quad 1 - \rho_{12}^2 \qquad \text{where} \quad \rho_{12} = \rho_{21}$$

Another way of stating this is that the conditional distribution of

$$\frac{\left[\gamma_2 + \delta_2\, f_J(x_2') - \rho_{12}\{\gamma_1 + \delta_1\, f_1(x_1')\}\right]}{\sqrt{1-\rho_{12}^2}}$$

given x_1' is standard normal. Therefore, the conditional distribution of x_2', given x_1', is of the same Johnson system type, S_J, as the marginal distribution of x_2', but with γ_2 replaced by

$$\frac{\left[\gamma_2 - \rho_{12}\{\gamma_1 + \delta_1\, f_1(x_1')\}\right]}{\sqrt{1-\rho_{12}^2}}$$

and δ_2 replaced by

$$\frac{\delta_2}{\sqrt{1-\rho_{12}^2}}$$

The median regressions of x_2' on x_1' were studied by Elderton and Johnson because the means of the Johnson system equations are complex, when compared with the easy median equations. I spent time developing these conditional median equations for the trivariate and quadrivariate Johnson distributions and later realized that they would not be required for this research. To prevent the loss of these equations, I listed them in Appendix E, in hopes that they could assist with further research.

3.5.2.2 Conditional Trivariate Johnson Equations

We define a trivariate Johnson distribution of type S_{IJK} by requiring the joint distribution of $z_1 = \gamma_1 + \delta_1 f_I(x_1')$, $z_2 = \gamma_2 + \delta_2 f_J(x_2')$, and $z_3 = \gamma_3 + \delta_3 f_K(x_3')$ to be described by a standardized trivariate normal distribution. There are 64 different S_{IJK} distributions based on the four Johnson systems. Using the properties of the standardized trivariate normal distribution, the conditional trivariate Johnson equations can be developed. The conditional distribution of $z_3 = \gamma_3 + \delta_3 f_K(x_3')$, given z_1 and z_2, is also standard normally distributed. If we rearrange and partition the variable array and the correlation matrix as follows:

$$\mathbf{Z} = \begin{bmatrix} z_3 \\ \hline z_1 \\ z_2 \end{bmatrix} \qquad \text{and} \qquad \mathbf{R} = \left[\begin{array}{c:cc} 1 & \rho_{31} & \rho_{32} \\ \hdashline \rho_{13} & 1 & \rho_{12} \\ \rho_{23} & \rho_{21} & 1 \end{array}\right]$$

and use the expected value equation, $\mathbf{E}[\mathbf{Z}_1 \mid \mathbf{Z}_2] = \mathbf{R}_{12}\mathbf{R}_{22}^{-1}\mathbf{Z}_2$, then the expected value of z_3, given z_1 and z_2, can be calculated as follows:

$$\mathbf{E}(z_3 \mid z_1, z_2) = \begin{bmatrix} \rho_{31} & \rho_{32} \end{bmatrix} \begin{bmatrix} 1 & \rho_{12} \\ \rho_{21} & 1 \end{bmatrix}^{-1} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

which solves to:

$$\mathbf{E}(z_3 \mid z_1, z_2) = \left(\frac{\rho_{13} - \rho_{12}\,\rho_{23}}{1-\rho_{12}^2}\right) z_1 + \left(\frac{\rho_{23} - \rho_{12}\,\rho_{13}}{1-\rho_{12}^2}\right) z_2$$

If we apply the variance equation, $\mathbf{Var}[\mathbf{Z}_1 \mid \mathbf{Z}_2] = \mathbf{R}_{11} - \mathbf{R}_{12}\,\mathbf{R}_{22}^{-1}\,\mathbf{R}_{21}$, then the variance of z_3, given z_1 and z_2, can be calculated as follows:

$$\mathbf{Var}(z_3 \mid z_1, z_2) = 1 - \begin{bmatrix} \rho_{31} & \rho_{32} \end{bmatrix} \begin{bmatrix} 1 & \rho_{12} \\ \rho_{21} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \rho_{13} \\ \rho_{23} \end{bmatrix}$$

which solves to:

$$\mathbf{Var}(z_3 \mid z_1, z_2) = 1 - \frac{\rho_{13}^2 + \rho_{23}^2 - 2\rho_{12}\,\rho_{23}\,\rho_{13}}{1-\rho_{12}^2}$$

Another way of stating this is that the conditional distribution of

$$\frac{\left[\gamma_3 + \delta_3\, f_K(x_3') - \left(\frac{\rho_{13} - \rho_{12}\,\rho_{23}}{1-\rho_{12}^2}\right)\{\gamma_1 + \delta_1\, f_I(x_1')\} - \left(\frac{\rho_{23} - \rho_{12}\,\rho_{13}}{1-\rho_{12}^2}\right)\{\gamma_2 + \delta_2\, f_J(x_2')\}\right]}{\sqrt{1 - \frac{\rho_{13}^2 + \rho_{23}^2 - 2\rho_{12}\,\rho_{23}\,\rho_{13}}{1-\rho_{12}^2}}}$$

given x_1' and x_2' is standard normal. Therefore, the conditional distribution of x_3', given x_1' and x_2', is of the same Johnson system type, S_K, as the marginal distribution of x_3', but with γ_3 replaced by

$$\frac{\left[\gamma_3 - \left(\frac{\rho_{13} - \rho_{12}\,\rho_{23}}{1-\rho_{12}^2}\right)\{\gamma_1 + \delta_1\, f_I(x_1')\} - \left(\frac{\rho_{23} - \rho_{12}\,\rho_{13}}{1-\rho_{12}^2}\right)\{\gamma_2 + \delta_2\, f_J(x_2')\}\right]}{\sqrt{1 - \frac{\rho_{13}^2 + \rho_{23}^2 - 2\rho_{12}\,\rho_{23}\,\rho_{13}}{1-\rho_{12}^2}}}$$

and δ_3 replaced by

$$\frac{\delta_3}{\sqrt{1 - \frac{\rho_{13}^2 + \rho_{23}^2 - 2\rho_{12}\,\rho_{23}\,\rho_{13}}{1-\rho_{12}^2}}}$$

3.5.2.3 Conditional Quadrivariate Johnson Equations

We define a quadrivariate Johnson distribution of type S_{IJKM} by requiring the joint distribution of $z_1 = \gamma_1 + \delta_1 f_I(x'_1)$, $z_2 = \gamma_2 + \delta_2 f_J(x'_2)$, $z_3 = \gamma_3 + \delta_3 f_K(x'_3)$, and $z_4 = \gamma_4 + \delta_4 f_M(x'_4)$ to be described by a standardized quadrivariate normal distribution. There are 256 different S_{IJKM} distributions based on the four Johnson systems. Using the properties of the standardized quadrivariate normal distribution, the conditional quadrivariate Johnson equations can be developed. The conditional distribution of $z_4 = \gamma_4 + \delta_4 f_M(x'_4)$, given z_1, z_2, and z_3, is also normally distributed. If we rearrange and partition the variable array and the correlation matrix as follows:

$$\mathbf{Z} = \left[\begin{array}{c} z_4 \\ \hdashline z_1 \\ z_2 \\ z_3 \end{array}\right] \qquad \text{and} \qquad \mathbf{R} = \left[\begin{array}{c:ccc} 1 & \rho_{41} & \rho_{42} & \rho_{43} \\ \hdashline \rho_{14} & 1 & \rho_{12} & \rho_{13} \\ \rho_{24} & \rho_{21} & 1 & \rho_{23} \\ \rho_{34} & \rho_{31} & \rho_{32} & 1 \end{array}\right]$$

and use the expected value equation, $\mathbf{E}[\mathbf{Z}_1 \mid \mathbf{Z}_2] = \mathbf{R}_{12}\,\mathbf{R}_{22}^{-1}\,\mathbf{Z}_2$, then the expected value of z_4, given z_1, z_2, and z_3, can be calculated as follows:

$$\mathbf{E}(z_4 \mid z_1, z_2, z_3) = \begin{bmatrix} \rho_{41} & \rho_{42} & \rho_{43} \end{bmatrix} \begin{bmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{21} & 1 & \rho_{23} \\ \rho_{31} & \rho_{32} & 1 \end{bmatrix}^{-1} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

which solves to:

$$\begin{aligned} \mathbf{E}(z_4 \mid z_1, z_2, z_3) = {} & \left(\frac{\rho_{14}\left(1-\rho_{23}^2\right) + \rho_{24}\left(\rho_{13}\,\rho_{23} - \rho_{12}\right) + \rho_{34}\left(\rho_{12}\,\rho_{23} - \rho_{13}\right)}{1 + 2\rho_{12}\,\rho_{23}\,\rho_{13} - \rho_{12}^2 - \rho_{23}^2 - \rho_{13}^2} \right) z_1 \\ & + \left(\frac{\rho_{14}\left(\rho_{13}\,\rho_{23} - \rho_{12}\right) + \rho_{24}\left(1-\rho_{13}^2\right) + \rho_{34}\left(\rho_{12}\,\rho_{13} - \rho_{23}\right)}{1 + 2\rho_{12}\,\rho_{23}\,\rho_{13} - \rho_{12}^2 - \rho_{23}^2 - \rho_{13}^2} \right) z_2 \\ & + \left(\frac{\rho_{14}\left(\rho_{12}\,\rho_{23} - \rho_{13}\right) + \rho_{24}\left(\rho_{12}\,\rho_{13} - \rho_{23}\right) + \rho_{34}\left(1-\rho_{12}^2\right)}{1 + 2\rho_{12}\,\rho_{23}\,\rho_{13} - \rho_{12}^2 - \rho_{23}^2 - \rho_{13}^2} \right) z_3 \end{aligned}$$

If we apply the variance equation, $\mathbf{Var}\left[\mathbf{Z}_1 \mid \mathbf{Z}_2\right] = \mathbf{R}_{11} - \mathbf{R}_{12}\,\mathbf{R}_{22}^{-1}\,\mathbf{R}_{21}$, then the variance of z_4, given z_1, z_2, and z_3, can be calculated as follows:

$$\mathbf{Var}\left(z_4 \mid z_1, z_2, z_3\right) = 1 - \begin{bmatrix} \rho_{41} & \rho_{42} & \rho_{43} \end{bmatrix} \begin{bmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{21} & 1 & \rho_{23} \\ \rho_{31} & \rho_{32} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \rho_{14} \\ \rho_{24} \\ \rho_{34} \end{bmatrix}$$

which solves to:

$$\begin{aligned} \mathbf{Var}\left(z_4 \mid z_1, z_2, z_3\right) = 1 - \Bigg\{ & \rho_{14}\left(\frac{\rho_{14}\left(1-\rho_{23}^2\right)+\rho_{24}\left(\rho_{13}\,\rho_{23}-\rho_{12}\right)+\rho_{34}\left(\rho_{12}\,\rho_{23}-\rho_{13}\right)}{1+2\rho_{12}\,\rho_{23}\,\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right) \\ & + \rho_{24}\left(\frac{\rho_{14}\left(\rho_{13}\,\rho_{23}-\rho_{12}\right)+\rho_{24}\left(1-\rho_{13}^2\right)+\rho_{34}\left(\rho_{12}\,\rho_{13}-\rho_{23}\right)}{1+2\rho_{12}\,\rho_{23}\,\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right) \\ & + \rho_{34}\left(\frac{\rho_{14}\left(\rho_{12}\,\rho_{23}-\rho_{13}\right)+\rho_{24}\left(\rho_{12}\,\rho_{13}-\rho_{23}\right)+\rho_{34}\left(1-\rho_{12}^2\right)}{1+2\rho_{12}\,\rho_{23}\,\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\Bigg\} \end{aligned}$$

Another way of stating this is that the conditional distribution of

$$\frac{\left[\begin{aligned} & \gamma_4+\delta_4 f_M\left(x_4'\right)-\left(\frac{\rho_{14}\left(1-\rho_{23}^2\right)+\rho_{24}\left(\rho_{13}\,\rho_{23}-\rho_{12}\right)+\rho_{34}\left(\rho_{12}\,\rho_{23}-\rho_{13}\right)}{1+2\rho_{12}\,\rho_{23}\,\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\left\{\gamma_1+\delta_1 f_I\left(x_1'\right)\right\} \\ & -\left(\frac{\rho_{14}\left(\rho_{13}\,\rho_{23}-\rho_{12}\right)+\rho_{24}\left(1-\rho_{13}^2\right)+\rho_{34}\left(\rho_{12}\,\rho_{13}-\rho_{23}\right)}{1+2\rho_{12}\,\rho_{23}\,\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\left\{\gamma_2+\delta_2 f_J\left(x_2'\right)\right\} \\ & -\left(\frac{\rho_{14}\left(\rho_{12}\,\rho_{23}-\rho_{13}\right)+\rho_{24}\left(\rho_{12}\,\rho_{13}-\rho_{23}\right)+\rho_{34}\left(1-\rho_{12}^2\right)}{1+2\rho_{12}\,\rho_{23}\,\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\left\{\gamma_3+\delta_3 f_K\left(x_3'\right)\right\} \end{aligned}\right]}{\sqrt{\begin{aligned} 1-\Bigg\{ & \rho_{14}\left(\frac{\rho_{14}\left(1-\rho_{23}^2\right)+\rho_{24}\left(\rho_{13}\,\rho_{23}-\rho_{12}\right)+\rho_{34}\left(\rho_{12}\,\rho_{23}-\rho_{13}\right)}{1+2\rho_{12}\,\rho_{23}\,\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right) \\ & +\rho_{24}\left(\frac{\rho_{14}\left(\rho_{13}\,\rho_{23}-\rho_{12}\right)+\rho_{24}\left(1-\rho_{13}^2\right)+\rho_{34}\left(\rho_{12}\,\rho_{13}-\rho_{23}\right)}{1+2\rho_{12}\,\rho_{23}\,\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right) \\ & +\rho_{34}\left(\frac{\rho_{14}\left(\rho_{12}\,\rho_{23}-\rho_{13}\right)+\rho_{24}\left(\rho_{12}\,\rho_{13}-\rho_{23}\right)+\rho_{34}\left(1-\rho_{12}^2\right)}{1+2\rho_{12}\,\rho_{23}\,\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\Bigg\} \end{aligned}}}$$

given x'_1, x'_2 and x'_3 is standard normal. Therefore, as with the bivariate and trivariate distributions, the conditional distribution of x'_4, given x'_1, x'_2, and x'_3, is of the same Johnson system type, S_M, as the marginal distribution of x'_4, but with γ_4 replaced by

$$\frac{\left[\begin{aligned}&\gamma_4-\left(\frac{\rho_{14}\left(1-\rho_{23}^2\right)+\rho_{24}\left(\rho_{13}\rho_{23}-\rho_{12}\right)+\rho_{34}\left(\rho_{12}\rho_{23}-\rho_{13}\right)}{1+2\rho_{12}\rho_{23}\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\left\{\gamma_1+\delta_1 f_I\left(x'_1\right)\right\}\\&-\left(\frac{\rho_{14}\left(\rho_{13}\rho_{23}-\rho_{12}\right)+\rho_{24}\left(1-\rho_{13}^2\right)+\rho_{34}\left(\rho_{12}\rho_{13}-\rho_{23}\right)}{1+2\rho_{12}\rho_{23}\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\left\{\gamma_2+\delta_2 f_J\left(x'_2\right)\right\}\\&-\left(\frac{\rho_{14}\left(\rho_{12}\rho_{23}-\rho_{13}\right)+\rho_{24}\left(\rho_{12}\rho_{13}-\rho_{23}\right)+\rho_{34}\left(1-\rho_{12}^2\right)}{1+2\rho_{12}\rho_{23}\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\left\{\gamma_3+\delta_3 f_K\left(x'_3\right)\right\}\end{aligned}\right]}{\sqrt{\begin{aligned}&1-\left\{\rho_{14}\left(\frac{\rho_{14}\left(1-\rho_{23}^2\right)+\rho_{24}\left(\rho_{13}\rho_{23}-\rho_{12}\right)+\rho_{34}\left(\rho_{12}\rho_{23}-\rho_{13}\right)}{1+2\rho_{12}\rho_{23}\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\right.\\&+\rho_{24}\left(\frac{\rho_{14}\left(\rho_{13}\rho_{23}-\rho_{12}\right)+\rho_{24}\left(1-\rho_{13}^2\right)+\rho_{34}\left(\rho_{12}\rho_{13}-\rho_{23}\right)}{1+2\rho_{12}\rho_{23}\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\\&\left.+\rho_{34}\left(\frac{\rho_{14}\left(\rho_{12}\rho_{23}-\rho_{13}\right)+\rho_{24}\left(\rho_{12}\rho_{13}-\rho_{23}\right)+\rho_{34}\left(1-\rho_{12}^2\right)}{1+2\rho_{12}\rho_{23}\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\right\}\end{aligned}}}$$

and δ_4 replaced by

$$\frac{\delta_4}{\sqrt{\begin{aligned}&1-\left\{\rho_{14}\left(\frac{\rho_{14}\left(1-\rho_{23}^2\right)+\rho_{24}\left(\rho_{13}\rho_{23}-\rho_{12}\right)+\rho_{34}\left(\rho_{12}\rho_{23}-\rho_{13}\right)}{1+2\rho_{12}\rho_{23}\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\right.\\&+\rho_{24}\left(\frac{\rho_{14}\left(\rho_{13}\rho_{23}-\rho_{12}\right)+\rho_{24}\left(1-\rho_{13}^2\right)+\rho_{34}\left(\rho_{12}\rho_{13}-\rho_{23}\right)}{1+2\rho_{12}\rho_{23}\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\\&\left.+\rho_{34}\left(\frac{\rho_{14}\left(\rho_{12}\rho_{23}-\rho_{13}\right)+\rho_{24}\left(\rho_{12}\rho_{13}-\rho_{23}\right)+\rho_{34}\left(1-\rho_{12}^2\right)}{1+2\rho_{12}\rho_{23}\rho_{13}-\rho_{12}^2-\rho_{23}^2-\rho_{13}^2}\right)\right\}\end{aligned}}}$$

Although the scope of this research does not include more than four variables, the equations required for additional variables can be developed in the same method as was outlined above.

3.5.3 Reverse Johnson Transformations

The conditional equations which were presented in the previous sections, all require the value of x_1' to be given. This requirement is essentially stating a very basic assumption that the first marginal distribution, S_1, is treated as independent of all others in order to start the sample generation. Because x_1' is treated as independent, it can be generated from univariate simulation techniques. Recall that the general form of the Johnson transformation system was:

$$z = \gamma + \delta f\left(\frac{x - \xi}{\lambda}\right)$$

where z is distributed as the standard normal, N(0,1). Box and Muller (1958) present a popular normal variate generator:

$$R = \mu + \sigma\sqrt{-2\,ln(U_1)}\,cos(2\pi U_2)$$

where $R \sim N(\mu,\sigma^2)$ and U_1 and U_2 are independent uniform 0-1, U(0,1), variates. If a standard normal variate, R(0,1) is substituted for z_1 in the Johnson system equations, then the value of x_1 can be solved for. The equations required to solve for x_1 depend on the Johnson system which represents the distribution of x_1. These equations are presented as follows:

$$z_1 = R(0,1)$$

S_B system: $$x_1 = \lambda_1\left[1 + exp\left(\frac{\gamma_1 - z_1}{\delta_1}\right)\right]^{-1} + \xi_1$$

S_U system: $$x_1 = \lambda_1\, sinh\left(\frac{z_1 - \gamma_1}{\delta_1}\right) + \xi_1$$

S_L system: $\quad x_1 = exp\left(\frac{z_1 - \gamma_1}{\delta_1}\right) + \xi_1$

S_S system: $\quad x_1 = \xi_1 - exp\left(\frac{z_1 - \gamma_1}{\delta_1}\right)$

S_N system: $\quad x_1 = \frac{z_1 - \gamma_1}{\delta_1}$

After the value of z_1 has been generated, it can be used in the conditional Johnson equations to calculate the values of z_2, z_3, and z_4 from the following equations:

$$
\begin{aligned}
z_2 &= R(\mathbf{E}[z_2 \mid z_1], \mathbf{Var}[z_2 \mid z_1]) \\
z_3 &= R(\mathbf{E}[z_3 \mid z_1, z_2], \mathbf{Var}[z_3 \mid z_1, z_2]) \\
z_4 &= R(\mathbf{E}[z_4 \mid z_1, z_2, z_3], \mathbf{Var}[z_4 \mid z_1, z_2, z_3])
\end{aligned}
$$

S_B system: $\quad x_i = \lambda_i \left[1 + exp\left(\frac{\gamma_i - z_i}{\delta_i}\right)\right]^{-1} + \xi_i \qquad$ for i = 2, 3, 4

S_U system: $\quad x_i = \lambda_i \, sinh\left(\frac{z_i - \gamma_i}{\delta_i}\right) + \xi_i \qquad$ for i = 2, 3, 4

S_L system: $\quad x_i = exp\left(\frac{z_i - \gamma_i}{\delta_i}\right) + \xi_i \qquad$ for i = 2, 3, 4

S_S system: $\quad x_i = \xi_i - exp\left(\frac{z_i - \gamma_i}{\delta_i}\right) \qquad$ for i = 2, 3, 4

S_N system: $\quad x_i = \frac{z_i - \gamma_i}{\delta_i} \qquad$ for i = 2, 3, 4

The values of x_1, x_2, x_3, and x_4 are simulated variates from the original observed multivariate Johnson distribution, and represent one sample from that distribution. Using this technique, we can generate as many samples as we would like.

3.6 Multivariate Process Capability Indices, C_{pa} and MC_{pa}

3.6.1 Dimension Specifications

It was stated in Chapter 1, that one of the delimitations of this study was that only independent specifications would be considered. The literature review clearly shows that dependent and independent specifications are the subject of some controversy, as they apply to multivariate capability indices. The issue of handling dependent tolerances needs to be addressed, however, it is beyond the scope of this research.

The upper and lower specification limits, USL and LSL, of each dimension (or variable or distribution) are used to determine the proportion nonconforming and, therefore, must be given. Their values will be represented as follows:

$$LSL_i \text{ and } USL_i \qquad \text{where } i = 1,\ldots,p \qquad (1 \le p \le 4)$$

where i is the variable identifier, and p is the number of variables in the study.

3.6.2 Proportion Nonconforming, p_L, p_U, p^*, and Mp^*

The phrase "proportion nonconforming" simply means what proportion of the product does not conform to given specifications. Proportion is usually measured in either percentage (%) or parts per million (ppm). A multivariate process will have both proportion nonconforming for each marginal distribution (p_i^*) and proportion nonconforming for the multivariate distribution (Mp^*). The proportion nonconforming of each marginal distribution can be further clarified by identifying both the proportion of product which is smaller than the LSL (p_{Li}) and the proportion which is larger than the USL (p_{Ui}). The relationship can be stated as follows:

$$p_i^* = p_{Li} + p_{Ui} \qquad \text{where } i = 1,\ldots,p \qquad (1 \le p \le 4)$$

where the proportion nonconforming will be measured in ppm. When the multivariate proportion nonconforming, Mp*, is discussed, we must first define a nonconforming product as a product which has any of its variables considered nonconforming. Since this can represent only one variable or all four variables nonconforming, there is no direct relationship between p_i^* and Mp*. This lack-of-relationship is a problem which has plagued most multivariate capability indices. Stated simply, the value of Mp* does not tell us anything about the performance of the marginal distributions. However, this need only be considered a problem if changes are required to reduce the value of Mp*. The problem can be fairly easily dealt with by also providing the values of p_{Li}, p_{Ui}, and p_i^* for each marginal distribution. These values should give a mental picture of any problems that exist with the marginal distributions, as they affect the value of Mp*. The exact values of the univariate proportions nonconforming for the five Johnson systems can be calculated from our knowledge of the normal distribution and the Johnson system transformations. These calculations can be accomplished with the following equations:

S_N system:
$$p_{Li} = \Phi\{\gamma_i + \delta_i \, LSL_i\}$$
$$p_{Ui} = 1 - \Phi\{\gamma_i + \delta_i \, USL_i\}$$

S_L system:
$$p_{Li} = \begin{cases} \Phi\{\gamma_i + \delta_i \, ln(LSL_i - \xi_i)\} & \text{for } (\xi_i < LSL_i) \\ 0 & \text{all other} \end{cases}$$
$$p_{Ui} = \begin{cases} 1 - \Phi\{\gamma_i + \delta_i \, ln(USL_i - \xi_i)\} & \text{for } (\xi_i < USL_i) \\ 0 & \text{all other} \end{cases}$$

S_S system:
$$p_{Li} = \begin{cases} 1 - \Phi\{\gamma_i + \delta_i \, ln(\xi_i - LSL_i)\} & \text{for } (\xi_i > LSL_i) \\ 0 & \text{all other} \end{cases}$$
$$p_{Ui} = \begin{cases} \Phi\{\gamma_i + \delta_i \, ln(\xi_i - USL_i)\} & \text{for } (\xi_i > USL_i) \\ 0 & \text{all other} \end{cases}$$

S_U system:

$$p_{Li} = \Phi\left\{\gamma_i + \delta_i \, sinh^{-1}\left(\frac{LSL_i - \xi_i}{\lambda_i}\right)\right\}$$

$$p_{Ui} = 1 - \Phi\left\{\gamma_i + \delta_i \, sinh^{-1}\left(\frac{USL_i - \xi_i}{\lambda_i}\right)\right\}$$

S_B system:

$$p_{Li} = \begin{cases} \Phi\left\{\gamma_i + \delta_i \, ln\left(\dfrac{LSL_i - \xi_i}{\xi_i + \lambda_i - LSL_i}\right)\right\} & \text{for } (\xi_i < LSL_i < \xi_i + \lambda_i) \\ 0 & \text{all other} \end{cases}$$

$$p_{Ui} = \begin{cases} 1 - \Phi\left\{\gamma_i + \delta_i \, ln\left(\dfrac{USL_i - \xi_i}{\xi_i + \lambda_i - USL_i}\right)\right\} & \text{for } (\xi_i < USL_i < \xi_i + \lambda_i) \\ 0 & \text{all other} \end{cases}$$

When one multivariate sample is generated, consisting of a single measurement from each marginal distribution, a check is made to determine whether each measurement is conforming, and thus deciding the entire sample's conformance. As multiple samples are generated, the total number of nonconforming samples is tracked as a running sum. This nonconforming total can be divided by the total number of generated samples to give the proportion of samples nonconforming, Mp*. The proportion nonconforming can be multiplied by 100 to give the percentage nonconforming, or since parts per million is a popular metric, the proportion nonconforming can also be transformed into ppm's by multiplying by 1,000,000 (1×10^6). It is very important to recognize the role which measurement resolution plays during the simulation and the count of nonconforming samples. For example, if only 100 samples were generated, then the resolution of proportion nonconforming will be in multiples of 10,000 ppm's. The significance of this problem will be explained in the following section.

3.6.3 Capability Index Transformation

When it comes to a physical meaning of the capability index, there is only one choice for most persons in industry. Practically anyone that deals with capability in industry understands, to some extent, the physical meaning of the C_p index. It is due to this well known and understood capability index that any newly presented indices must have a similar physical meaning. Part of the understood physical meaning of the C_p index is the direct relationship between the index value and the potential proportion of nonconforming product. This relationship is defined by Littig et al. (1992) as: $p = 2[1-\Phi(3C_p)]$, where $\Phi(x)$ is the cumulative standardized normal distribution. The proposed univariate process capability index, C_{pa}, and multivariate process capability index, MC_{pn}, are transformations on the proportion of nonconforming product and are defined as follows:

$$C_{pn_i} = \frac{1}{3}\Phi^{-1}\left(1 - \frac{p_i^*}{2}\right) \qquad \text{where } i = 1,\ldots,p \qquad (1 \le p \le 4)$$

And

$$MC_{pn} = \frac{1}{3}\Phi^{-1}\left(1 - \frac{Mp^*}{2}\right) \qquad \text{where } p_i^* > 0 \text{ and } Mp^* > 0$$

where $\Phi^{-1}(x)$ is the inverse of the cumulative standardized normal distribution. This definition maintains the well-known C_p-relationship and allows the index value to be equal to one when the proportion nonconforming is equal to 2,700 ppm. This is the point at which the measurement resolution problem, discussed briefly in the previous section, becomes significant. The resolution of proportion nonconforming creates the resolution of the capability index. The previous example generated 100 samples, where the minimum applicable proportion nonconforming would be 0.01 or 10,000 ppm. The calculated (maximum) index of this proportion nonconforming would be 0.859 with only 100 possible incremental values achievable. This situation is most likely unacceptable to most practitioners, and the issue becomes a trade-off between computation time and resolution.

The program that is attached, which will be discussed in the last chapter, generates one million samples, where the minimum applicable proportion nonconforming would be 0.000001 or 1 ppm. The calculated (maximum) index of this proportion nonconforming is 1.689. For most practitioners, this is an acceptable maximum value for the capability index, primarily because the index is used to indicate process performance problems. This value would indicate that the process is performing very well and probably doesn't rate any expensive engineering resources, when compared to others. The value also represents the limits of the inverse cumulative standard normal function which was utilized in the program. Another important issue occurs when the generated sample contains no samples outside of specifications, i.e., when the proportion nonconforming is calculated to be zero. This situation can occur when either the process performance is extremely good, the specification(s) lie outside of a bounded distribution, or possibly by the error of estimating tail probability by counting a fixed number of generated samples, when the process performance is very good. Checks are placed in the program code preventing the calculation of the capability index, which theoretically would be equal to infinity. The value of the capability index is set equal to 2.0, which allows its identification in the program output.

3.6.4 Confidence Intervals

The topic of confidence intervals on capability indices is interesting to me, primarily since I have not seen their use in industry. Kotz and Johnson (1993) present a fairly detailed review of the current literature that exists on this topic. Since the C_p index is only a function of the standard deviation of a process, the confidence intervals for the C_p index are based directly upon the confidence intervals for the standard deviation. If the normal distribution is assumed, the formula for the $100(1 - \alpha)\%$ confidence interval on the standard deviation is as follows:

$$s\sqrt{\frac{n-1}{\chi^2_{\alpha/2,\,n-1}}} \;<\; \sigma \;<\; s\sqrt{\frac{n-1}{\chi^2_{1-\alpha/2,\,n-1}}}$$

where $\chi^2_{1-\alpha,\upsilon}$ represents the cumulative chi-square distribution with υ degrees of freedom. Construction of confidence intervals on the C_{pk} index is more difficult due to the index being a function of both the process standard deviation and the process mean. If the normal distribution is assumed, the formula for the $100(1 - \alpha)\%$ confidence interval on the mean is as follows:

$$\bar{x} - t_{\alpha/2,\,n-1}\frac{s}{\sqrt{n}} \;<\; \mu \;<\; \bar{x} + t_{\alpha/2,\,n-1}\frac{s}{\sqrt{n}}$$

where $t_{1-\alpha,\upsilon}$ represents the cumulative t-distribution with υ degrees of freedom. Kotz and Johnson caution readers on the approach of calculating confidence intervals on both process parameters, and then using the results to calculate confidence intervals on the C_{pk} index. There are a number of issues with this approach which were presented and need to be discussed in this research. The confidence interval on C_{pk} would represent the minimum and maximum possible values for C_{pk} corresponding to pairs of values $(\mu\ ,\sigma)$

within the rectangular region defined by the separate confidence intervals of both parameters.

The first point which was brought up was that if each of the parameter confidence intervals were set up with a $100(1 - \alpha)\%$ confidence coefficient, then the resulting confidence interval on C_{pk} will not, in general, be a $100(1 - \alpha)\%$ confidence interval. If we assumed that the parameter confidence intervals were mutually exclusive, then the resulting simultaneous confidence coefficient would be $100(1 - \alpha)^2\ \%$. This is to say that if 95% confidence intervals were constructed on both parameters, then we would only have 90.25% confidence that the intervals simultaneously contain both parameters. The second point which was brought up is that only part of the confidence rectangle contributes to the confidence interval for C_{pk}, and that there are other pairs of values (μ, σ) outside the rectangular region which could give C_{pk} values in the same interval. This point is said to counterbalance the effect of the first point, however, to an unknown, likely lesser, degree.

3.6.4.1 Univariate Confidence Intervals

If the four-parameter Johnson system is considered, we can quickly lose ourselves in complexity thinking of the ramification of simultaneous confidence intervals on four parameters. However, if we consider that the Johnson systems are transformations to the standard normal distribution, we can use some engineering judgement and quickly see that γ and δ are simple functions of the mean and standard deviation of the transformed normal distribution. These two parameters could then probably be considered as the primary contributors to the confidence interval of the capability index. We can apply the same confidence interval equations to these parameters of the Johnson system. The formulas are as follows:

$$\hat{\delta}\sqrt{\frac{\chi^2_{1-\alpha/2,n-1}}{n-1}} \quad < \quad \delta \quad < \quad \hat{\delta}\sqrt{\frac{\chi^2_{\alpha/2,n-1}}{n-1}}$$

and

$$\hat{\gamma}-\frac{t_{\alpha/2,n-1}}{\sqrt{n}} \quad < \quad \gamma \quad < \quad \hat{\gamma}+\frac{t_{\alpha/2,n-1}}{\sqrt{n}} \qquad\qquad \text{where } s \equiv \sigma$$

Note that the statistically invalid assumption, $s \equiv \sigma$, was used to create the confidence interval on γ. However, I think it is obvious that exact confidence intervals on the Johnson systems are either impossible or beyond the scope of this research. It is the attempt of this research to create "useable" confidence intervals which, as a minimum, will give the practitioner some useful indication of the accuracy of estimation on the capability index.

Using similar logic as presented previously, the confidence interval on each C_{pa} index would represent the minimum and maximum possible values for C_{pa} corresponding to pairs of values (γ , δ) within the rectangular region defined by the separate confidence intervals of both parameters on each distribution. Although it would seem possible to maximize and minimize the index calculation through optimization of the γ and δ parameters within their respective confidence intervals, some experimentation has lead to a much simpler approach. If we consider that the confidence rectangle is made from the upper and lower confidence interval limits on both parameters, and the parameter estimates represent a point in the central region of the rectangle, then we have identified nine possible (γ , δ) pairs that represent the perimeter and center of the confidence rectangle. The rectangle and identified points are displayed as Figure 3.7.

Graphical (γ , δ) Point Identification

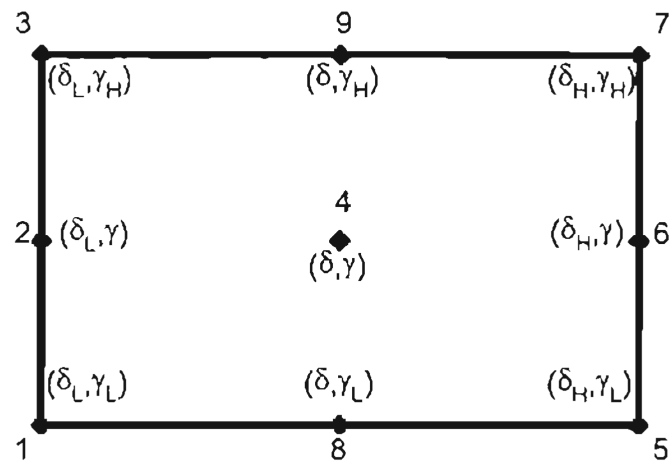


Figure 3.7: C_{pa} Confidence Rectangle on the (γ , δ) Plane

Through simulation studies and evaluation of these nine points, it was determined that the minimum value of C_{pa} was calculated using one of the three points on the left perimeter of the rectangle, labeled as points 1, 2, and 3. The maximum value of C_{pa} was calculated using one of the three points on the right perimeter of the rectangle, labeled as points 5, 6, and 7. Therefore, lower confidence limits for both p* and C_{pa} can be estimated by calculating their values at the three specified (γ , δ) pairs and selecting the minimum values. Similarly, the upper confidence limits for p* and C_{pa} can be estimated by calculating their values at the other three specified (γ , δ) pairs and selecting the maximum values. These values are used to approximate the 100(1 - α)% confidence limits on the univariate proportion nonconforming, p*, and the univariate capability index, C_{pa}, for each marginal distribution. For the purpose of consistent output, an α

value of 0.05 was used for these univariate calculations, attempting to achieve 95% confidence limits.

3.6.4.2 Multivariate Confidence Intervals

Recalling that the multivariate proportion nonconforming, Mp*, was estimated through the use of simulation, we are left with a similar situation when estimating confidence intervals for Mp* and the multivariate process capability index, MC_{pn}. My first attempt at calculating a confidence limit on Mp* consisted of using same (γ , δ) pairs that were selected for the univariate confidence limits. However, the resulting confidence intervals were extremely large and had no practical use. It became apparent that the α value which was used to calculate the univariate confidence limits would have to be adjusted to compensate for the complex relationship between the multivariate and univariate proportions nonconforming. However, when the α value changes, then so does the confidence rectangle shown in Figure 3.7. It would seem logical that the rectangle positions of the (γ , δ) pairs, resulting in the minimum and maximum values of p* and C_{pn}, would remain constant if everything else were held constant. Therefore, even though the actual values of γ and δ for each marginal distribution would change, their positions on their respective confidence rectangles would remain the same. Since we are merely attempting to approximate the 95% confidence limits on the multivariate proportion nonconforming, Mp*, and the multivariate capability index, MC_{pn}, the assumption is made that the combination of these new (γ , δ) pairs for each marginal distribution will yield the overall minimums and maximums on the theoretical multivariate confidence region. Although it would seem probable that the α value would need to be adjusted based on the number of marginal distributions, a constant α value of 0.20 was used in all multivariate calculations. This α value was selected based on the results of simulation studies attempting to achieve 95% confidence intervals, primarily with the S_N system. The end result is that a million samples are generated in the program code to approximate each of the multivariate confidence limits, as well as the capability index point estimate, discussed previously. The penalty of estimating these confidence limits resides in the computational time required to generate an additional two million multivariate Johnson samples.

3.7 Summary

The methodologies developed in this chapter allow one to calculate a multivariate process capability index and an approximate confidence interval by fitting the marginal distributions with the Johnson transformation systems, calculating the correlation between the transformed marginal distributions, simulating the original multivariate Johnson distribution, calculating the proportion nonconforming, and transforming that value into an index which has physical meaning.

The results and discussion of the proposed process capability indices are presented in the next chapter.

CHAPTER IV

RESULTS AND DISCUSSION

4.1 Multivariate Correlated Standard Normal Generator Performance

Because the multivariate correlated standard normal generator represents the backbone of this research, a high level of confidence must be achieved concerning its performance. To test the generator's performance, a quadrivariate standard normal distribution was generated, with correlation coefficients between each variable specified to be 0.6. The results of the test are shown below in Table 4.1:

Table 4.1
Multivariate Correlated Standard Normal Generator Performance on a S_{NNNN} Johnson System (Sample Size = 10,000)

Marginal Distributions	Actual		Calculated	
	μ	σ	$\bar{x}$	s
$N_1(0,1,1,0)$	0	1	-0.0095	1.0006
$N_2(0,1,1,0)$	0	1	-0.0073	0.9971
$N_3(0,1,1,0)$	0	1	-0.0136	1.0003
$N_4(0,1,1,0)$	0	1	-0.0097	1.0026
Correlations	**Actual**		**Calculated**	
ρ_{12}	0.6		0.6010	
ρ_{13}	0.6		0.6023	
ρ_{23}	0.6		0.6018	
ρ_{14}	0.6		0.5971	
ρ_{24}	0.6		0.5996	
ρ_{34}	0.6		0.6005	

Recall that when correlation is discussed in this study, it is always referring to the correlation between the Z-values of the variables. That is to say that the variables have been transformed to the standard normal before the correlation is calculated. In this test case, the variables were all standard normal to begin with, thus there would be no difference in the correlation coefficients after the transformations. Also note the unique nomenclature of the listed variables, where $N_i(\xi,\lambda,\delta,\gamma)$ represents a normal, S_N, Johnson system with the Johnson parameters listed inside the parenthesis. The i represents the variable number. This parameter order inside the parenthesis will be maintained throughout this study.

The results shown in Table 4.1 show good performance and demonstrate a high confidence level with respect to accuracy. The author has no explanation for the calculated means being all negative. The appearance of bias is most likely the result of random chance.

4.2 Effects of Correlation, Number of Variables, and Process Mean Shifts on the Multivariate Process Capability Index, MC_{pn}

A basic assumption of this research is that both correlation and number of variables affect the value of the multivariate process capability index or the proportion of nonconforming product. To test the validity of this assumption, a number of strategic test cases were run to demonstrate the effects of correlation, number of variables, and process mean shifts on the multivariate process capability index, MC_{pn}, and indirectly on the multivariate proportion of nonconforming product. The tests are based on the basic understanding that the univariate distribution, $S_N(0,1,1,0)$ represents a standard normal distribution, and when the lower and upper specification limits are set to –3 and 3, respectively, the resulting proportion nonconforming will be 2,700 PPM, representing a univariate process capability index value of 1.0. The $S_N(0,1,1,-1)$ is a normal distribution with $\sigma = 1.0$ and $\mu = 1.0$, representing a mean shift of one standard deviation to the right. The results of these tests are shown below in Table 4.2:

Table 4.2
Effects of Correlation, Number of Variables, and Process Mean Shifts on the Multivariate Process Capability Index, MC_{pa}

Actual Johnson System	Spec. Limits (Low , High)	Average Actual MC_{pa}	
		$\rho_{ij} = 0.0$	$\rho_{ij} = 0.6$
$S_{NN}(0,1,1,0)$	(-3 , 3)	0.9274	0.9331
$S_{NN}(0,1,1,-1)$	(-3 , 3)	0.6681	0.6843
$S_{NNN}(0,1,1,0)$	(-3 , 3)	0.8826	0.8939
$S_{NNN}(0,1,1,-1)$	(-3 , 3)	0.6111	0.6416
$S_{NNNN}(0,1,1,0)$	(-3 , 3)	0.8497	0.8659
$S_{NNNN}(0,1,1,-1)$	(-3 , 3)	0.5688	0.6122

Note that this research does not present exact formulas to calculate the multivariate proportion nonconforming. It is for this reason that simulation techniques are employed to estimate their values. The column heading of Average Actual MC_{pa}, in Table 4.2, was calculated by inputting the exact parameters into the multivariate process capability program and allowing it to estimate the multivariate proportion nonconforming through simulation. The only source of error in this approach is the simulation error caused by estimating a proportion from a finite number of generated samples. A number of trials were performed, with the average of the results being considered the "actual" value. During this process it was discovered that the multivariate proportion nonconforming of the centered distributions with this level of univariate proportion nonconforming (approximately equal to or less than 2,700 PPM) could be calculated by summing the univariate proportions nonconforming, which can be calculated using equations presented in the previous chapter. However, these requirements are placed on all variables, and thus, are likely valuable only in this presentation example.

The results shown in Table 4.2 should be analyzed between sources of change. The univariate standard normal, $S_N(0,1,1,0)$, has a capability of 1.0, as presented earlier. The bivariate, trivariate, and quadrivariate standard normals have capabilities of 0.927, 0.883, and 0.850, respectively. This demonstrates that as the number of variables increases, the capability decreases, when everything else is held constant. When correlation between variables is introduced, the bivariate, trivariate, and quadrivariate standard normals have capabilities of 0.933, 0.894, and 0.866, respectively. This demonstrates that when correlation exists the capabilities increase slightly. This phenomena can be partially explained by the discussion following Figure 1.5 in Chapter 1. This discussion can be summarized by saying that when correlation exists, the probability of a sample being nonconforming on more than one set of specifications, increases. However, when a sample is nonconforming on more than one set of specifications, it is only counted as nonconforming once for the total multivariate proportion nonconforming. This yields a smaller multivariate proportion nonconforming, resulting in the increases in capability.

The variables with process mean shifts were presented to show the combined effects of correlation with process mean shifts. To demonstrate this phenomena, we recognize that the presence of correlation on the bivariate standard normal caused a capability increase of 0.006. However, correlation on the bivariate shifted standard normal caused a capability increase of 0.016, which represents an approximate tripling of the capability increase. The trivariate and quadrivariate distributions show similar results. The reasons for this phenomena can also be partially explained by similar logic as before. This can be summarized by saying that the probability of a sample being nonconforming on more that one set of specifications, further increases when one or more of the processes is not centered. This increased probability causes further decreases in the total multivariate proportion nonconforming, resulting in larger increases in the multivariate capability.

4.3 Johnson System Selection Performance

It is very intuitive that if this research is based upon fitting a Johnson system to an actual distribution, a high level of confidence that the selected system's properties are close to the actual system's properties, is very important. When this occurs, the capability calculations will represent as close to the truth as possible. This situation does not exist in this form when using the normal assumption. When all variables are assumed to be normally distributed, the error caused by a non-normal distribution is built into the original assumption. The problem of calling a normally distributed variable non-normal does not exist. Unfortunately, this problem does exist in this research, and it is for this reason that the system selection decision matrix, presented in Chapter 3, is critical to the success of this study. To demonstrate the ability of the proposed system selection decision matrix, a number of univariate test cases, with numerous repetitive trials, were performed with process sample sizes set at both 30 and 100. These sample sizes were chosen due to their realistic nature when industrial application is considered. The results of these test are tabulated below in Table 4.3:

Table 4.3
System Selection Performance Based on Actual Johnson Systems

Actual Johnson System	Process Sample Size	Qty. of Trials	Selected Johnson System (Percentage of Trials)				
			S_N	S_L	S_S	S_B	S_U
$S_N(0,1,1,0)$	30	7,388	94.73%	1.85%	1.69%	1.65%	0.08%
$S_N(0,1,1,0)$	100	9,720	93.25%	2.34%	2.22%	1.69%	0.50%
$S_L(0,1,1,0)$	30	486	3.29%	73.66%	0.00%	23.05%	0.00%
$S_L(0,1,1,0)$	100	425	0.00%	58.12%	0.00%	41.88%	0.00%
$S_S(0,1,1,0)$	30	504	3.57%	0.00%	72.82%	23.61%	0.00%
$S_S(0,1,1,0)$	100	502	0.00%	0.00%	41.63%	58.37%	0.00%
$S_B(0,1,0.5,0)$	30	1,000	93.20%	0.20%	0.10%	6.50%	0.00%
$S_B(0,1,0.5,0)$	100	289	41.56%	0.00%	0.35%	58.13%	0.00%
$S_B(0,1,0.25,0)$	30	532	50.00%	0.38%	0.00%	49.62%	0.00%
$S_B(0,1,0.25,0)$	100	482	7.26%	0.41%	0.00%	92.33%	0.00%
$S_U(0,1,1,0)$	30	532	50.94%	22.74%	16.54%	5.08%	4.70%
$S_U(0,1,1,0)$	100	438	10.96%	14.16%	14.38%	9.82%	50.68%

The results, shown in Table 4.3, of particular interest have been highlighted. These results represent the percentage of trials when the selected system was the same as the actual system. It can be seen that when the actual variable is standard normally distributed, then the decision matrix yields an approximate 95% probability of selecting the normal system from the sample's descriptive statistics. The author believes that this is the most important result, as it minimizes the error discussed previously. The reason for this result's importance, if not readily apparent, will be discussed in the following paragraph.

Of the five Johnson systems presented, only two do not have boundaries; the normal and the unbounded systems. The importance of this observation is based on how the capability indices are statistically calculated. The systems without boundaries have tails which extend to theoretical infinity. This causes there to always be a certain proportion nonconforming outside of any specification limits, no matter how far they are away from the process mean. The systems which have boundaries; lognormal (bounded on the left), special (bounded on the right), and bounded (bounded on both the right and the left), do not have tails beyond their boundaries. Thus, when a given specification limit resides outside of a variable's boundary, then there will be no proportion nonconforming on that particular side of the distribution. This situation is certainly acceptable when it exists in reality, however, if in reality the distribution does have a tail, then certain obvious problems can occur when calculating or estimating the proportion nonconforming. The most likely result of a system with boundaries being selected when the actual process does not have a boundary where the selected system does, is that the proportion nonconforming will be underestimated, causing a higher capability index value than actually exists. This is the reason why the primary objective of the decision matrix is to select a system without boundaries, primarily the normal system, when the actual process does not have boundaries. Note that when the reverse situation occurs, when a system without boundaries is selected for a process which actually has boundaries, then the capability estimate is conservative. This, of course, is not nearly as bad of a situation.

The actual system's parameters play a key role in the performance of the system selection decision matrix. This is demonstrated with the bounded system trials. It can be seen that the $S_B(0,1,0.25,0)$ distribution trials were much more likely to have the S_B system selected than those trials from the $S_B(0,1,0.5,0)$ distribution. This result is due to the fact that the $S_B(0,1,0.5,0)$ distribution is much less normally-shaped than the $S_B(0,1,0.25,0)$ distribution. The fact remains that each of the non-normal Johnson systems have sets of parameters which allow them to be shaped very similarly to the normal distribution. The end result is that when a sample comes from a non-normal distribution, the farther the shape of that distribution is from the normal-shape, the probability increases that a non-normal system will be selected to represent it.

4.4 Performance of the Univariate Process Capability Index, C_{pu}

The performance of the univariate process capability index, C_{pn}, is demonstrated by again selecting test cases with known parameters, specification limits, and thus capabilities. The results are tabulated below in Table 4.4.

Table 4.4
Performance of the Univariate Process Capability Index, C_{pn}, and its Confidence Intervals

Actual Johnson System	Spec. Limits (Low , High)	Process Sample Size	Actual C_{pa}	Qty. of Trials	Average Calculated C_{pa}	Percent within C.I.	Average C.I. Half-Width
S_N(0,1,1,0)	(-3 , 3)	30	1.000	7,388	1.0385	96.18%	0.3166
S_N(0,1,1,0)	(-3 , 3)	100	1.000	9,720	1.0177	94.93%	0.1672
S_L(0,1,1,0)	(0.01 , 16.15)	30	1.000	286	0.7076	30.77%	0.1807
S_L(0,1,1,0)	(0.01 , 16.15)	100	1.000	220	1.1602	53.18%	0.2168
S_L(0,1,1,0)	(0.3256 , 16.15)	30	0.500	200	0.4796	61.50%	0.1080
S_L(0,1,1,0)	(0.3256 , 16.15)	100	0.500	205	0.6952	58.05%	0.1477
S_S(0,1,1,0)	(-16.15 , -0.01)	30	1.000	304	0.6964	31.25%	0.1759
S_S(0,1,1,0)	(-16.15 , -0.01)	100	1.000	302	1.0343	45.03%	0.1821
S_S(0,1,1,0)	(-16.15 , -0.3256)	30	0.500	200	0.4844	70.50%	0.1232
S_S(0,1,1,0)	(-16.15 , -0.3256)	100	0.500	200	0.4912	97.50%	0.1239
S_B(0,1,0.5,0)	(0.00245 , 0.9975)	30	1.000	1,000	0.5483	1.90%	0.1873
S_B(0,1,0.5,0)	(0.00245 , 0.9975)	100	1.000	289	0.7662	24.22%	0.1220
S_B(0,1,0.25,0)	(6.1E-6 , 0.9999938)	30	1.000	332	0.3678	0.00%	0.1355
S_B(0,1,0.25,0)	(6.1E-6 , 0.9999938)	100	1.000	186	0.4545	0.00%	0.0724
S_B(0,1,0.25,0)	(0.00247 , 0.99753)	30	0.500	200	0.3998	83.50%	0.1369
S_B(0,1,0.25,0)	(0.00247 , 0.99753)	100	0.500	296	0.4335	82.09%	0.0689
S_U(0,1,1,0)	(-10.02 , 10.02)	30	1.000	154	1.6535	42.21%	0.3667
S_U(0,1,1,0)	(-10.02 , 10.02)	100	1.000	132	1.4232	52.27%	0.3091
S_U(0,1,1,0)	(-2.13 , 2.13)	30	0.500	378	0.5120	84.13%	0.2178
S_U(0,1,1,0)	(-2.13 , 2.13)	100	0.500	306	0.5150	74.18%	0.1334

Note that the system parameters and the specification limits were selected to have actual univariate capabilities of either 1.0 or 0.5. This was done purposely to show that the performance differs under different levels of capability. Also note that the last column of values represents the average half-width of the confidence interval. This value was determined by subtracting the lower confidence level from the upper confidence level and dividing by two. It is an attempt to show the plus or minus width of the confidence interval, even though the confidence intervals are not necessarily symmetrical.

As in the previous section, the results, shown in Table 4.4, of particular interest are the results of the standard normal distribution trials. It can be seen that both the performance of the C_{pn} index and its confidence interval for the standard normal distributions are very satisfactory. Also recognize that the calculated values of the capability index are slightly higher than the actual values. This is caused by the small percentage of trials which were fitted by systems with boundaries, causing smaller proportions nonconforming, and thus slightly elevated capability index values. The remaining test cases have a variety of performance levels. This can again be attributed somewhat to parameter selection. As with the previous system selection section, the system parameters also play a key role in the performance of both the capability index and its confidence intervals. Of particular interest is the performance between those systems with an actual capability of 1.0 versus those with an actual capability of 0.5. With a close look at the results in Table 4.4, one can see the increase in performance as the capability of the actual distribution decreases. Again, the reason for this phenomena is that when the capability of the actual distribution decreases, the proportion nonconforming is based less on the tails of the distribution. This means that even if a system with boundaries was selected when the actual system did not have a boundary, then the chances of the specification limits falling inside the boundary increases. When this occurs, the errors in calculating the proportion nonconforming decrease, yielding better performance of the univariate capability index, C_{pa}.

4.5 Performance of the Multivariate Process Capability Index, MC_{pa}

As in previous sections, test cases with known parameters and specification limits were used to demonstrate the performance of the Multivariate Capability Index, MC_{pa}. The results are tabulated in Table 4.5, which can be found on a following page. The same univariate distributions, tabulated in Table 4.4, were used in the multivariate distributions, tabulated in Table 4.5. One of the reasons for presenting the data in this manner was so the changes in the capability could be recognized as the number of variables increased. With the exception of the trivariate and quadrivariate normal test cases, the test cases are all bivariate distributions with identical marginal distributions. The author found no reasons to present mixed system multivariate distributions. The knowledge that they would have been just as easy to select as any other non-mixed system is what is important.

Table 4.5
Performance of the Multivariate Process Capability Index, MC_{pa}, and its Confidence Intervals

Actual Johnson System	Spec. Limits (Low , High)	Process Sample Size	Average Actual MC_{pa}	Qty. of Trials	Average Calculated MC_{pa}	Percent within C.I.	Average C.I. Half-Width
$S_{NN}(0,1,1,0)$	(-3 , 3)	30	0.9274	134	0.9189	94.03%	0.2075
$S_{NN}(0,1,1,0)$	(-3 , 3)	100	0.9274	500	0.9193	94.60%	0.1241
$S_{NNN}(0,1,1,0)$	(-3 , 3)	30	0.8826	200	0.8659	96.50%	0.2155
$S_{NNN}(0,1,1,0)$	(-3 , 3)	100	0.8826	400	0.8717	95.25%	0.1328
$S_{NNNN}(0,1,1,0)$	(-3 , 3)	30	0.8497	200	0.8177	97.50%	0.2222
$S_{NNNN}(0,1,1,0)$	(-3 , 3)	100	0.8497	488	0.8421	97.34%	0.1262
$S_{LL}(0,1,1,0)$	(0.01 , 16.15)	30	0.9274	143	0.5638	6.99%	0.1063
$S_{LL}(0,1,1,0)$	(0.01 , 16.15)	100	0.9274	110	0.9367	38.18%	0.1389
$S_{LL}(0,1,1,0)$	(0.3256 , 16.15)	30	0.3839	100	0.3571	65.00%	0.0724
$S_{LL}(0,1,1,0)$	(0.3256 , 16.15)	100	0.3839	100	0.5056	63.00%	0.0975
$S_{SS}(0,1,1,0)$	(-16.15 , -0.01)	30	0.9274	152	0.5635	7.24%	0.1030
$S_{SS}(0,1,1,0)$	(-16.15 , -0.01)	100	0.9274	151	0.7901	31.79%	0.1151
$S_{SS}(0,1,1,0)$	(-16.15 , -0.3256)	30	0.3839	100	0.3639	68.00%	0.0817
$S_{SS}(0,1,1,0)$	(-16.15 , -0.3256)	100	0.3839	100	0.3732	99.00%	0.0878
$S_{BB}(0,1,0.5,0)$	(0.00245 , 0.9975)	30	0.9274	500	0.4310	0.00%	0.1272
$S_{BB}(0,1,0.5,0)$	(0.00245 , 0.9975)	100	0.9274	103	0.5877	6.80%	0.0772
$S_{BB}(0,1,0.25,0)$	(6.1E-6 , 0.9999938)	30	0.9274	166	0.3079	0.00%	0.0962
$S_{BB}(0,1,0.25,0)$	(6.1E-6 , 0.9999938)	100	0.9274	93	0.3791	0.00%	0.0537
$S_{BB}(0,1,0.25,0)$	(0.00247 , 0.99753)	30	0.3839	100	0.3021	46.00%	0.0962
$S_{BB}(0,1,0.25,0)$	(0.00247 , 0.99753)	100	0.3839	148	0.3548	76.35%	0.0511
$S_{UU}(0,1,1,0)$	(-10.02 , 10.02)	30	0.9274	77	1.4291	59.74%	0.3227
$S_{UU}(0,1,1,0)$	(-10.02 , 10.02)	100	0.9274	66	1.1628	46.97%	0.2479
$S_{UU}(0,1,1,0)$	(-2.13 , 2.13)	30	0.3839	189	0.3808	82.54%	0.1566
$S_{UU}(0,1,1,0)$	(-2.13 , 2.13)	100	0.3839	153	0.3887	64.05%	0.1022

As in the previous sections, the multivariate standard normal results, shown in Table 4.5, are of particular interest. It can be seen that both the performance of the MC_{pa} index and its confidence interval for the multivariate standard normal distributions are very satisfactory. Also recognize that the MC_{pa} estimates are almost always conservative to the actual values. It is of some comfort to know that when error exists in the approximation that it usually shows up as biased towards the conservative side of the capability index estimate. As in the univariate test cases, parameter selection is very important when reviewing the performance of the multivariate capability index, MC_{pn}. The performance can also be seen to increase when the actual process capability decreases. The same explanations presented in the previous section, also pertain to this section.

4.6 Selected Bivariate Case Study Examples

As a method of verification of the performance of this research, a case study presented in other literature was utilized. Wang et al. (1996) utilize data which they label as "Sultan (1986) bivariate processes data". This data consists of two variables: H (the Brinell hardness) and S (the tensile strength). The data set consists of 25 samples. Although the actual data set is not presented by Wang et al. (1996), the data set can be found in Chan et al. (1991). This data set is shown on the following page as Table 4.6.

Wang et al. (1996) present two examples utilizing this hardness / strength data set. The first example set the lower and upper specifications on H as 112.7 and 241.3, respectively, and the lower and upper specification on S as 32.7 and 73.3, respectively. The second example set the lower and upper specifications on H as 86.15 and 214.75, respectively, and the lower and upper specifications on S as 24.75 and 65.35, respectively. They use three different methods to calculate the bivariate process capability as discussed in the literature review. The third method is the probability of nonconforming product method presented by Chen (1994). This approach to multivariate capability falls into the same category as the approach presented in this research, therefore, it will be used to compare results. Wang et al. (1996) also present their results under three different confidence levels: 99.73%, 99%, and 95%. The 99.73% confidence level results are utilized for comparison, due to this percentage represents the six-sigma natural tolerance utilized by this research.

Table 4.6
Hardness / Strength Case Study Data Set

Brinell Hardness (H)	Tensile Strength (S)
143	34.2
200	57.0
160	47.5
181	53.4
148	47.8
178	51.5
162	45.9
215	59.1
161	48.4
141	47.3
175	57.3
187	58.5
187	58.2
186	57.0
172	49.4
182	57.2
177	50.6
204	55.1
178	50.9
196	57.9
160	45.5
183	53.9
179	51.2
194	57.5
181	55.6

They show the results of the first example, using method 3 and 99.73% confidence level as 1.12, and for the second example as 0.81. The output produced by the proposed multivariate capability index program for the first example is shown below in Table 4.7. The output for the second example is shown in Table 4.8.

Table 4.7
Hardness / Strength Case Study Example # 1 Output

UNIVARIATE STATISTICS	VARIABLE # 1	VARIABLE # 2
Selected Johnson Distribution	N	S
NS (Number of Samples)	25	25
LSL (Lower Spec. Limit)	+112.700000	+32.700000
USL (Upper Spec. Limit)	+241.300000	+73.300000
E (Xi)	0.000000	+67.525853
L (Lambda)	+1.000000	+1.000000
D (Delta)	+0.055514	+2.862306
G (Gamma)	-9.837160	-7.610644
F-Value of K-S Test	95.99%	42.79%
PL (PPM < LSL)	171	5,362
PU (PPM > USL)	187	0
P* (Total PPM Out-of-Spec.)	358	5,362
P* - Lower Confidence Limit	12,633	0
P* - Upper Confidence Limit	732,626	763,785
Cpa (Capability Index)	1.189765	0.928145
Cpa - Lower Confidence Limit	0.113878	0.100171
Cpa - Upper Confidence Limit	2.000000	2.000000

MULTIVARIATE STATISTICS	SYSTEM	CORRELATION COEFFICIENT P12
Multivariate Johnson System	NS	-0.847283
MP* (Total PPM Out-of-Spec.)	5,558	
MP* - Lower Confidence Limit	3,008	
MP* - Upper Confidence Limit	602,407	
MCpa (Capability Index)	0.924265	
MCpa - Lower Confidence Limit	0.173648	
MCpa - Upper Confidence Limit	0.988997	

Table 4.8
Hardness / Strength Case Study Example # 2 Output

UNIVARIATE STATISTICS	VARIABLE # 1	VARIABLE # 2
Selected Johnson Distribution	N	S
NS (Number of Samples)	25	25
LSL (Lower Spec. Limit)	+86.150000	+24.750000
USL (Upper Spec. Limit)	+214.750000	+65.350000
E (Xi)	0.000000	+67.525853
L (Lambda)	+1.000000	+1.000000
D (Delta)	+0.055514	+2.862306
G (Gamma)	-9.837160	-7.610644
F-Value of K-S Test	95.99%	42.79%
PL (PPM < LSL)	0	845
PU (PPM > USL)	18,554	0
P* (Total PPM Out-of-Spec.)	18,554	845
P* - Lower Confidence Limit	19	0
P* - Upper Confidence Limit	953,547	616,186
Cpa (Capability Index)	0.784785	1.112542
Cpa - Lower Confidence Limit	0.019418	0.167088
Cpa - Upper Confidence Limit	1.428028	2.000000

MULTIVARIATE STATISTICS	SYSTEM	CORRELATION COEFFICIENT P12
Multivariate Johnson System	NS	
MP* (Total PPM Out-of-Spec.)	19,344	-0.847283
MP* - Lower Confidence Limit	46	
MP* - Upper Confidence Limit	860,666	
MCpa (Capability Index)	0.779613	
MCpa - Lower Confidence Limit	0.058509	
MCpa - Upper Confidence Limit	1.358179	

The result for the first example is $MC_{pa} = 0.92$, and for the second example is $MC_{pa} = 0.78$. Although they are slightly conservative when compared with 1.12 and 0.81, they are not too far off. The differences can be attributed to the fact that the Johnson special system, S_S, was selected to fit the tensile strength (S) distribution, while the Chen (1994) approach assumed multivariate normality. The reason for the special system selection ended up being caused by the Z-value of the standard error of γ_1, Z_{γ_1}, which was calculated to be 2.39 for the tensile strength distribution. Since this value was greater than 1.96, we rejected the null hypothesis that this level of skew could have been caused by sampling error, even though the F-value for the normal fit was much better.

To further validate this research, the author forced the Johnson normal system to be selected on both the H and S variables, with the output for the first example shown in Table 4.9, and the second example in Table 4.10. The result for the first example is MC_{pa} = 1.14, and for the second example is MC_{pa} = 0.76. The result of the first example is much closer to the 1.12 value presented by Wang et al. (1996).

Table 4.9
Hardness / Strength Case Study Example # 1 Output (Forced S_N System)

UNIVARIATE STATISTICS	VARIABLE # 1	VARIABLE # 2
Selected Johnson Distribution	N	N
NS (Number of Samples)	25	25
LSL (Lower Spec. Limit)	+112.700000	+32.700000
USL (Upper Spec. Limit)	+241.300000	+73.300000
E (Xi)	0.000000	0.000000
L (Lambda)	+1.000000	+1.000000
D (Delta)	+0.055514	+0.176009
G (Gamma)	-9.837160	-9.208089
F-Value of K-S Test	95.99%	85.00%
PL (PPM < LSL)	171	278
PU (PPM > USL)	187	111
P* (Total PPM Out-of-Spec.)	358	388
P* - Lower Confidence Limit	12,633	12,229
P* - Upper Confidence Limit	732,626	635,678
Cpa (Capability Index)	1.189765	1.182780
Cpa - Lower Confidence Limit	0.113878	0.157917
Cpa - Upper Confidence Limit	2.000000	2.000000

MULTIVARIATE STATISTICS	SYSTEM	CORRELATION COEFFICIENT P12
Multivariate Johnson System	NN	
MP* (Total PPM Out-of-Spec.)	633	
MP* - Lower Confidence Limit	5,397	+0.833830
MP* - Upper Confidence Limit	273,207	
MCpa (Capability Index)	1.139027	
MCpa - Lower Confidence Limit	0.365236	
MCpa - Upper Confidence Limit	1.912818	

Table 4.10
Hardness / Strength Case Study Example # 2 Output (Forced S_N System)

UNIVARIATE STATISTICS	VARIABLE # 1	VARIABLE # 2
Selected Johnson Distribution	N	N
NS (Number of Samples)	25	25
LSL (Lower Spec. Limit)	+86.150000	+24.750000
USL (Upper Spec. Limit)	+214.750000	+65.350000
E (Xi)	0.000000	0.000000
L (Lambda)	+1.000000	+1.000000
D (Delta)	+0.055514	+0.176009
G (Gamma)	-9.837160	-9.208089
F-Value of K-S Test	95.99%	85.00%
PL (PPM < LSL)	0	1
PU (PPM > USL)	18,554	10,892
P* (Total PPM Out-of-Spec.)	18,554	10,893
P* - Lower Confidence Limit	19	27
P* - Upper Confidence Limit	953,547	911,935
Cpa (Capability Index)	0.784785	0.848704
Cpa - Lower Confidence Limit	0.019418	0.036866
Cpa - Upper Confidence Limit	1.428028	1.400088

MULTIVARIATE STATISTICS	SYSTEM	CORRELATION COEFFICIENT P12
Multivariate Johnson System	NN	
MP* (Total PPM Out-of-Spec.)	23,394	
MP* - Lower Confidence Limit	66	+0.833830
MP* - Upper Confidence Limit	725,817	
MCpa (Capability Index)	0.755645	
MCpa - Lower Confidence Limit	0.116898	
MCpa - Upper Confidence Limit	1.330239	

As a final check on the program, the variable order was switched on the second example. When compared to the results in Table 4.8, the univariate statistics were identical and the multivariate statistics were within an acceptable level of variation. Again, this variation is due to the proportion nonconforming estimate coming from a finite number of generated samples.

CHAPTER V

PROGRAM OPERATION

5.1 Running the Database Application

The program which is attached to this research was written in Microsoft Visual Basic version 5.0, under Microsoft Access version 97. This was done primarily to utilize some of the complex mathematical functions contained in Microsoft Excel version 97. Using these functions within Microsoft Access is much easier than with a stand-alone copy of Microsoft Visual Basic version 5.0. To run the database application, the user's computer must have both Microsoft Excel and Access, of version 97 or higher, installed on the computer. The application is called Thesis1.mdb, and is located on the 3.5" floppy disk attached to the back cover of this research.

When the application is started, a start-up form is launched. A graphic display of this form can be seen on Figure 5.1. This form, as first presented, has three command buttons which can be clicked by the user. These buttons are the Calculate Process Capability Index on Sample button (Capability button, for short), the Generate Sample from Johnson System button (Generate button, for short), and the Cancel button. If the Cancel button is clicked on this form now, the database application will close.

Start-Up Form

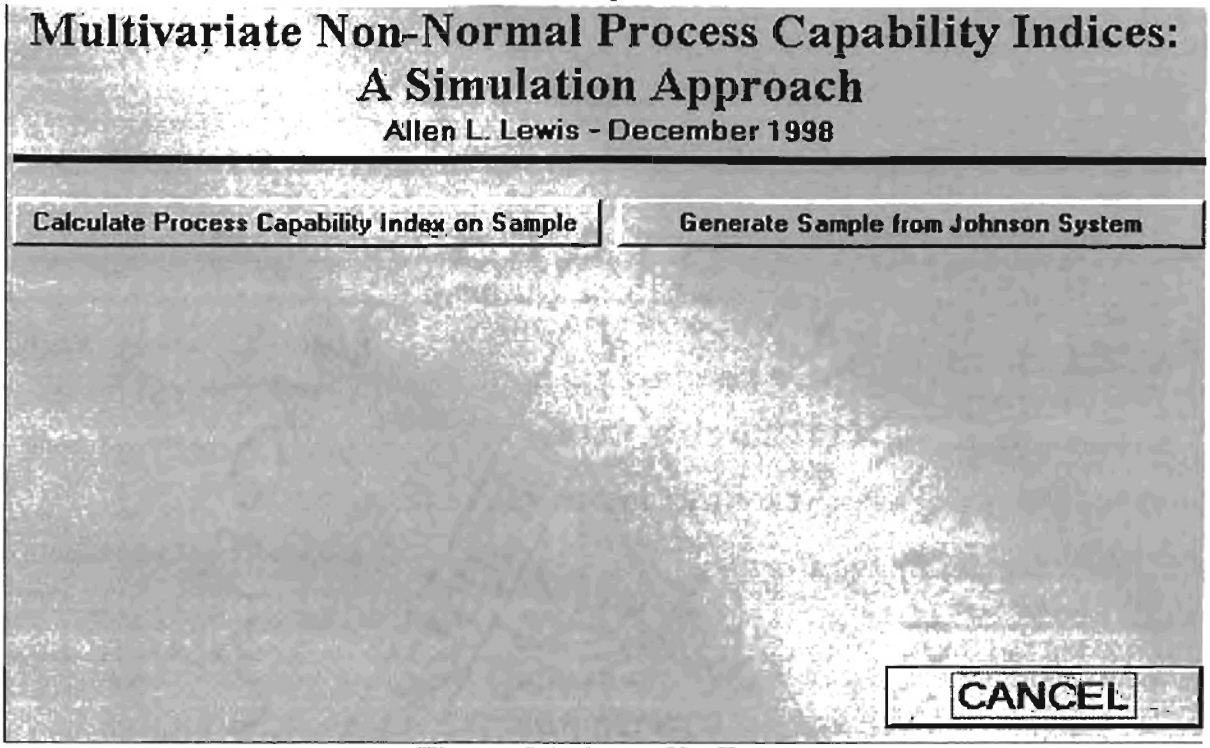


Figure 5.1: Start-Up Form

If the Capability button is clicked, more buttons and information appear on the form. Figure 5.2 shows a graphical display of this form after the Capability button has been clicked. The primary concern to the user at this stage is the accuracy of the input and output file specifications. The specifications at start-up are either the default settings of the computer or the settings stored by the previous user. In either case, the settings can be adjusted by selecting one of the "Change" buttons on the right side of the form. The Input File button and the Output File button are hyperlinks to the input file and output file listed on the screen. This is a quick way of viewing those files without leaving the application. If the Cancel button were clicked on this form now, the form would change back to the start-up form, displayed in Figure 5.1.

Start-Up Form (after Capability button click)

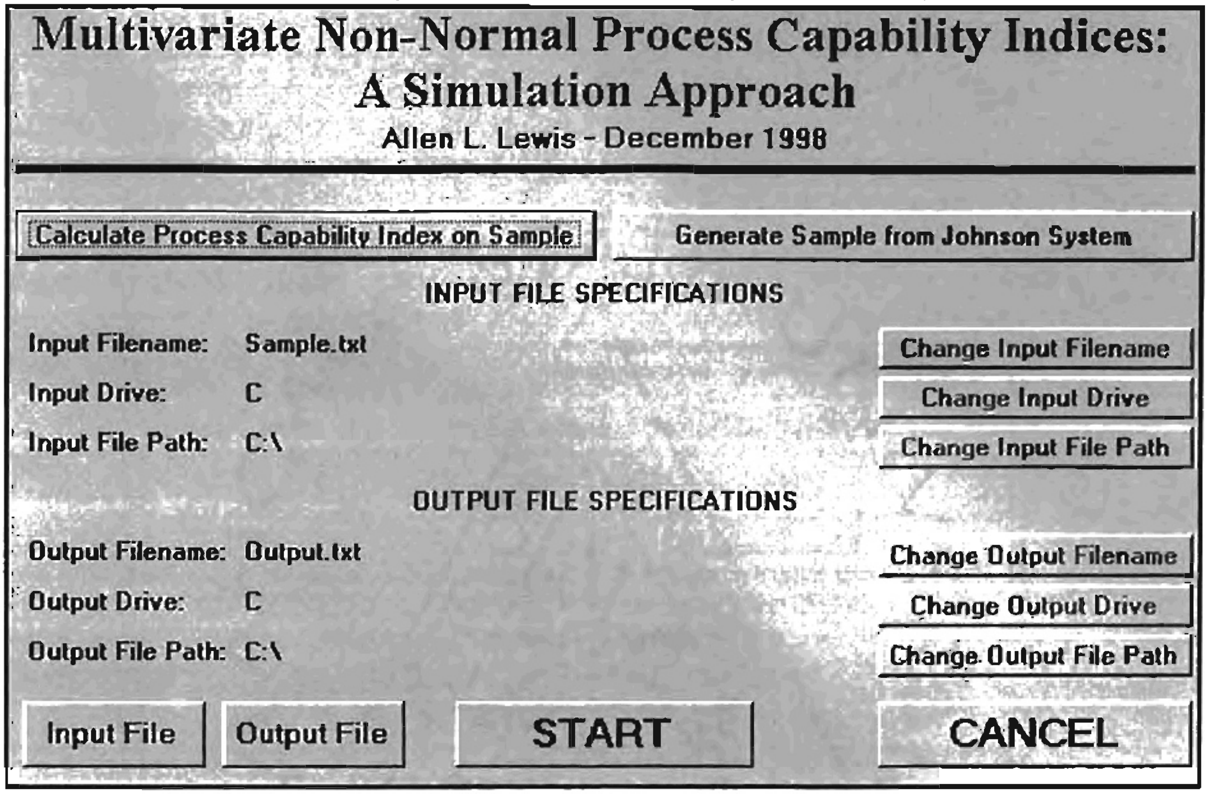


Figure 5.2: Start-Up Form (after Capability button click)

When the input and output file specifications are correct, and the input file exists in the correct format, the Start button can be clicked to start the program running. When the program is finished, a message box is displayed telling the user that the data was written to the output file. A graphical display of this form and message box is shown as Figure 5.3. When the OK button is clicked on the message box, the start-up form is again displayed, except that the hyperlink Input File and Output File buttons still appear. The output file can then be viewed from within the application. Of course this file can be copied and pasted into other documents, such as Microsoft Word documents.

Start-Up Form (with message box)

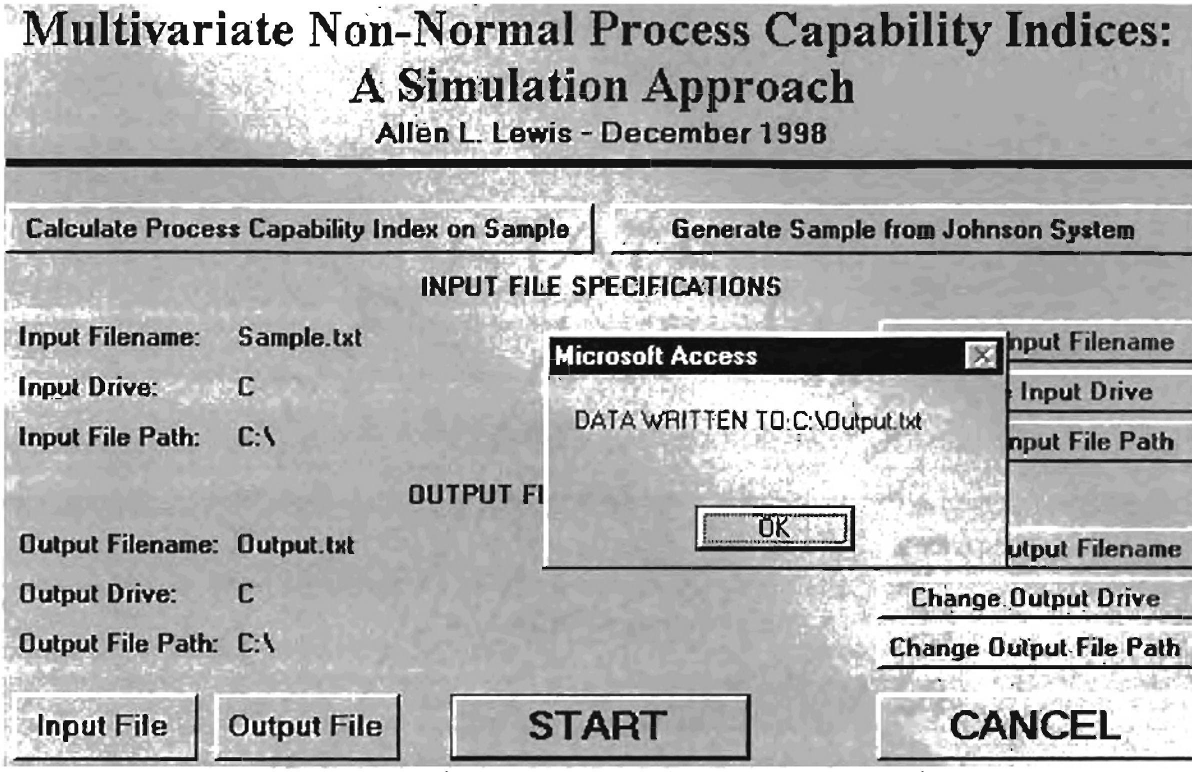


Figure 5.3: Start-Up Form (with message box)

If, instead of clicking the Capability button, the Generate button were clicked, then a different set of information and buttons appear. Figure 5.4 shows a graphical display of this form after the Generate button has been clicked. The primary concern to the user at this stage is the accuracy of the output file specifications. Again, these specifications at start-up are either the default settings of the computer or the settings stored by the previous user. As before, these settings can be adjusted by clicking the "Change" buttons on the right side of the form. If the Cancel button were clicked on this form now, the form would change back to the start-up form, displayed in Figure 5.1.

Start-Up Form (after Generate button click)

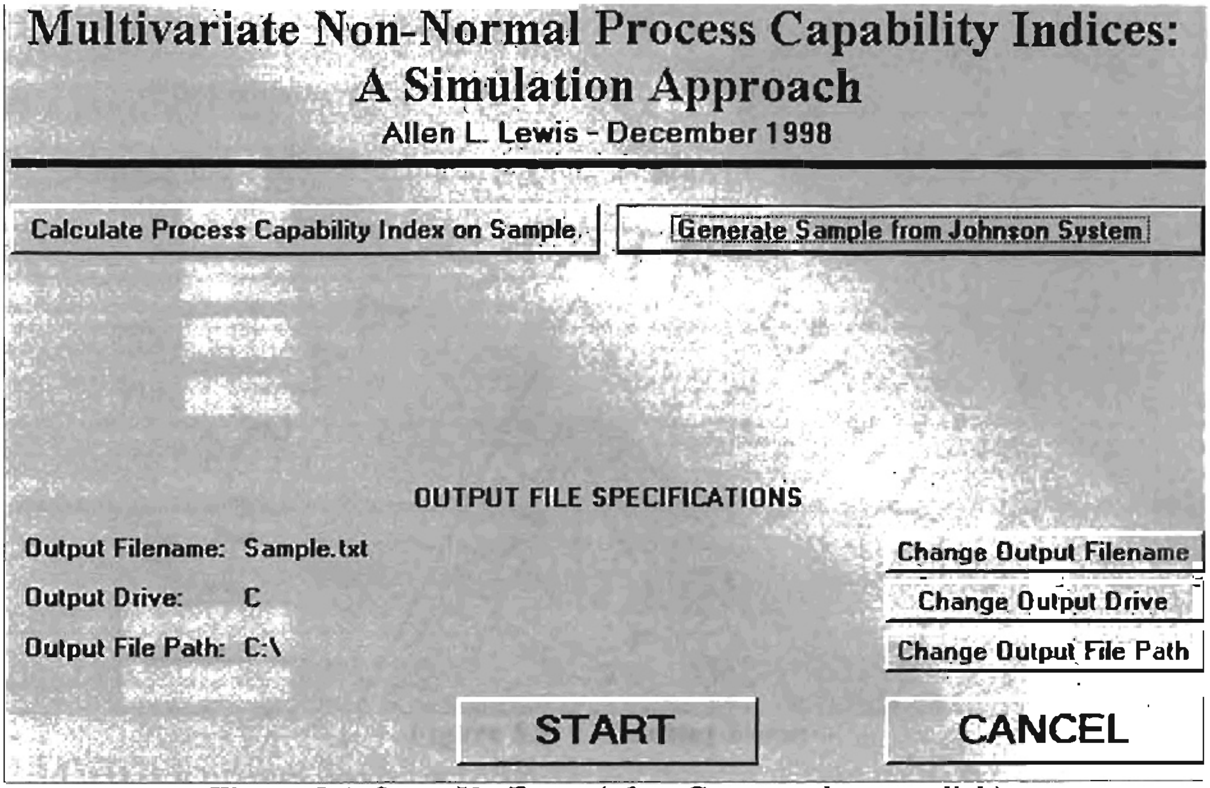


Figure 5.4: Start-Up Form (after Generate button click)

When the output file specifications are correct, the Start button can be clicked to open the Generator Form. A graphical display of this form is shown as Figure 5.5.

Generator Form

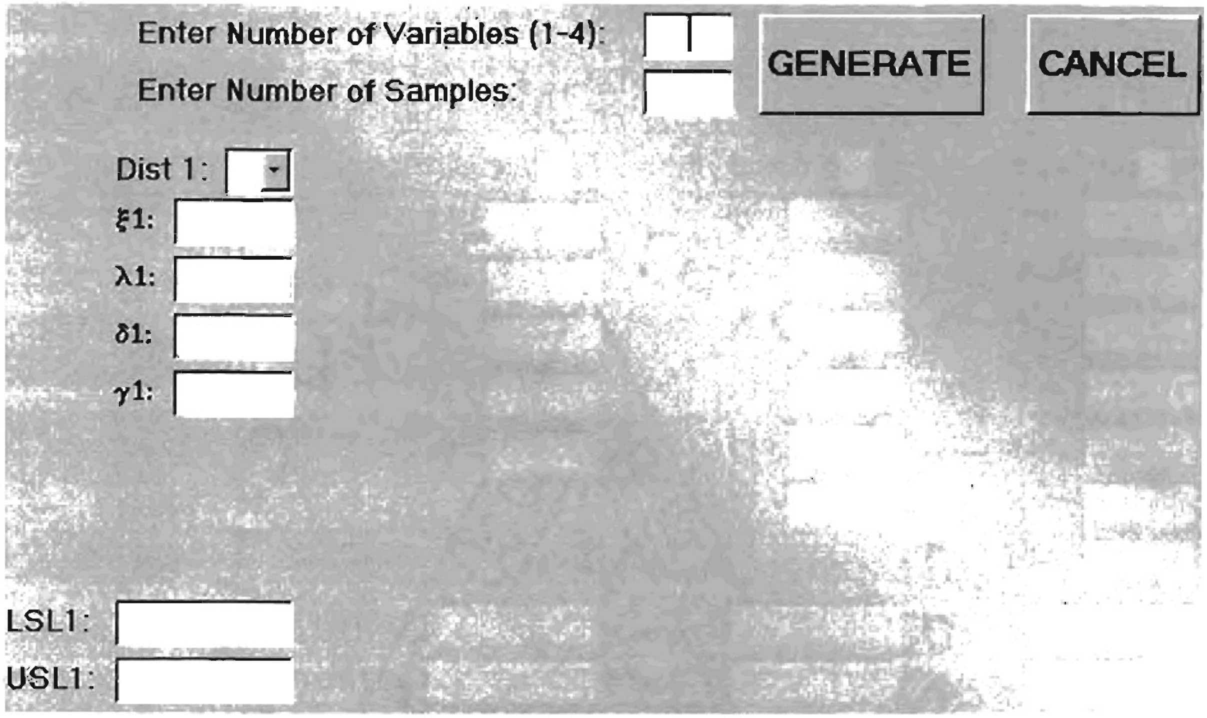


Figure 5.5: Generator Form

If the Cancel button were clicked on this form, it would close and bring the start-up form back to the screen. The first entry to be made on this form is the number of variables desired in the sample, which is limited to four in this study. When the entry is made and the Enter key is depressed, the appropriate number of entry boxes appear on the form. After the number of samples desired is input into the second entry box, the cursor jumps to the first distribution selection box. This box has a drop-down box which lists the valid selections. Figure 5.6 shows a graphical display of the form at this stage.

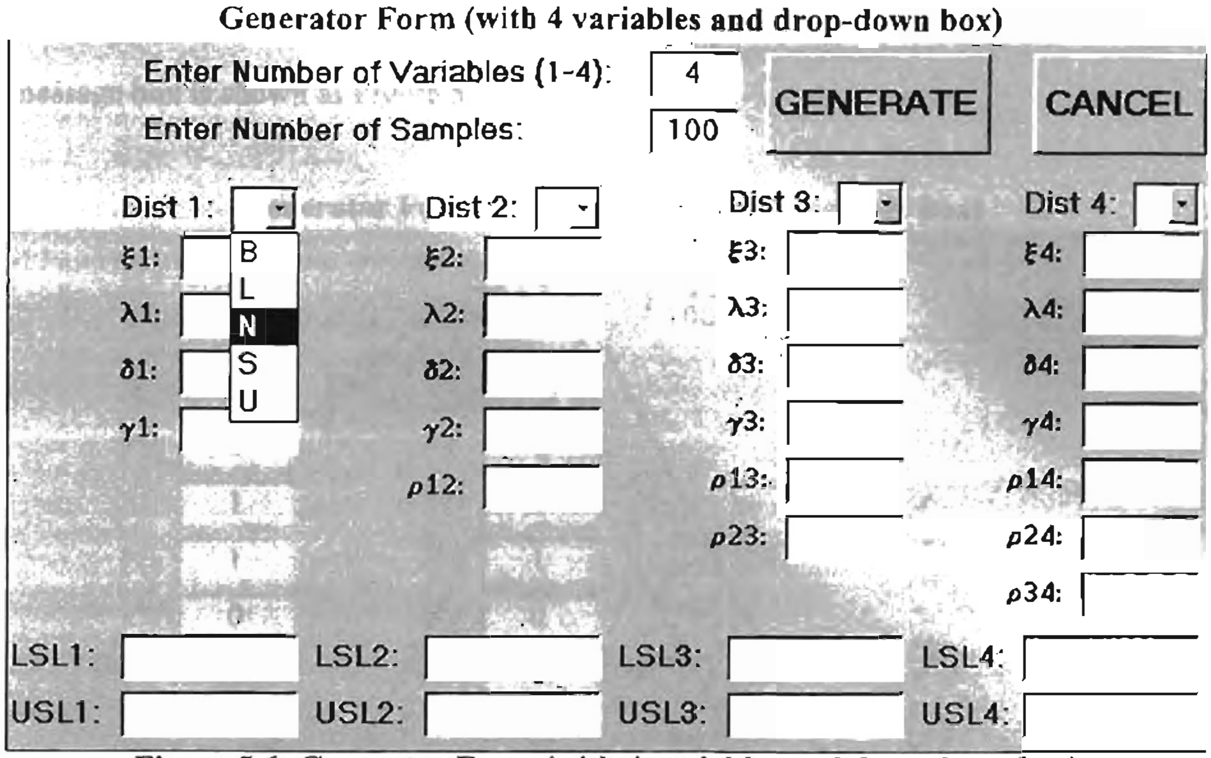


Figure 5.6: Generator Form (with 4 variables and drop-down box)

The valid selections for each distribution are "B" for the Johnson Bounded System, S_B, "L" for the Johnson Lognormal System, S_L, "N" for the Johnson Normal System, S_N, "S" for the Johnson-Lewis Special System, S_S, and "U" for the Johnson Unbounded System, S_U. The values of the four Johnson system parameters are the next entries to be made. Note that the S_N, S_L, and S_S systems do not require all four parameters. If one of those systems are selected, the unneeded parameter entry boxes will not be available. The last required entries for each variable are the correlation coefficients, after transformation, and the specification limits desired on the variable. These values are required to calculate a capability index. When all entries have been made, the Generate button can be clicked to create the output file with the newly generated samples. A message box will appear telling the user that the data was written to the output file. A graphical display of this form (with two variables selected) and message box is shown as Figure 5.7.

Generator Form (with 2 variables and message box)

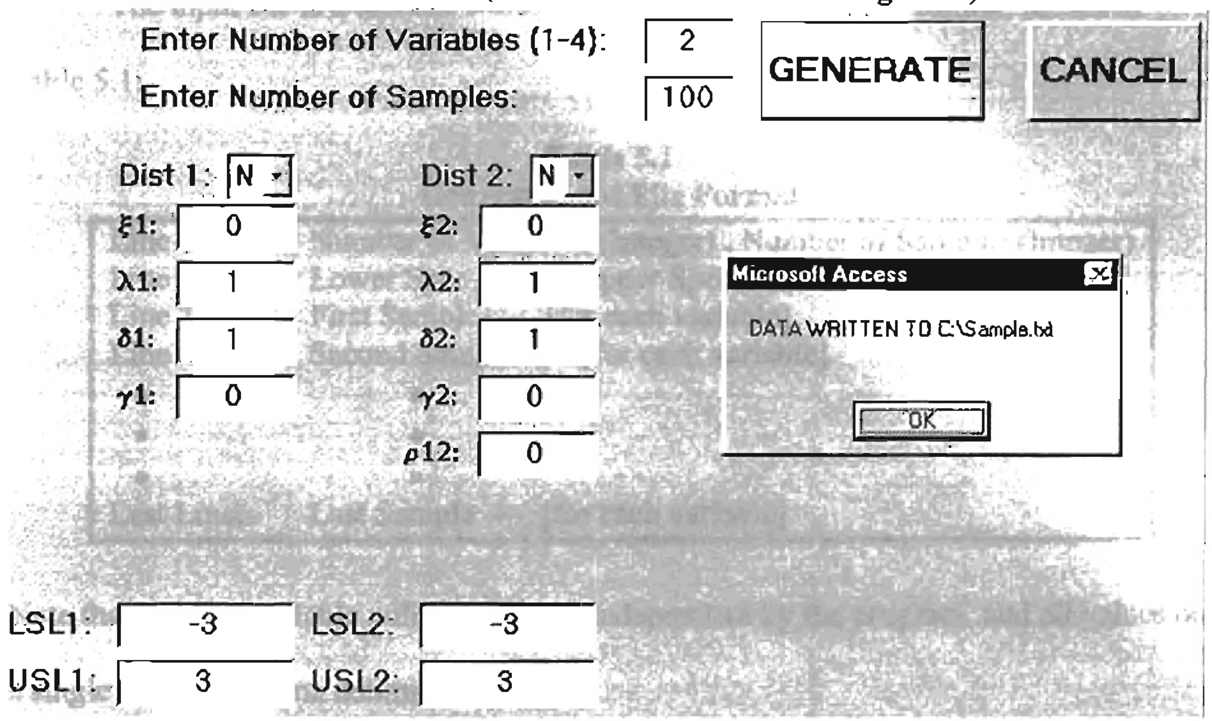


Figure 5.7: Generator Form (with 2 variables and message box)

When the OK button is clicked on the message box, the start-up form is again displayed. It is of value to note that if the user's desire is to run the newly generated sample on the capability program, then the user should set the Generator Form's output file name to be the same as the Capability Form's input file name. If this is done and the Capability button is clicked off the start-up form, then the contents of the generated sample can be viewed by clicking the hyperlink Input File button. Note that the output of the Generator Form will always be in the required format of the input file for the capability program.

5.2 Input File Format

5.2.1 General Input File Format

The input file is a text file which has the following required format, shown in Table 5.1:

Table 5.1
General Input File Format

Line 1 :	Number of Variables (Integer) , Number of Samples (Integer)
Line 2 :	Lower Spec Limit , Upper Spec Limit --- [for each variable]
Line 3 :	First Sample --- [for each variable]
Line 4 :	Second Sample --- [for each variable]
•	•
•	•
•	•
Last Line:	Last Sample --- [for each variable]

Note that the first variable will be assumed independent by the program, and all values on a single line are comma delimited.

5.2.2 Univariate Input File Example

A simple example of a univariate input file is shown below in Table 5.2:

Table 5.2
Univariate Input File Example

1, 10
20, 30
22.3
27.6
25.8
21.5
26.6
29.0
23.8
28.2
21.4
25.7

5.2.3 Bivariate Input File Example

The input file which was used in the first bivariate sample case example presented in Chapter 4 (Hardness / Strength data set) is shown below as Table 5.3:

Table 5.3
Hardness / Strength Case Study Example # 1 Input

```
2,25
112.7, 241.3, 32.7, 73.3
143, 34.2
200, 57.0
160, 47.5
181, 53.4
148, 47.8
178, 51.5
162, 45.9
215, 59.1
161, 48.4
141, 47.3
175, 57.3
187, 58.5
187, 58.2
186, 57.0
172, 49.4
182, 57.2
177, 50.6
204, 55.1
178, 50.9
196, 57.9
160, 45.5
183, 53.9
179, 51.2
194, 57.5
181, 55.6
```

5.3 Output File Format

5.3.1 Univariate Output File Example

A general univariate output file example is shown below as Table 5.4.

Table 5.4
Univariate Output File Example

```
    UNIVARIATE STATISTICS              VARIABLE # 1
------------------------------         ------------
Selected Johnson Distribution               N
NS (Number of Samples)                  100
LSL (Lower Spec. Limit)                -3.000000
USL (Upper Spec. Limit)                +3.000000
E (Xi)                                  0.000000
L (Lambda)                             +1.000000
D (Delta)                              +1.024142
G (Gamma)                              -0.170114
F-Value of K-S Test                     81.27%
PL (PPM < LSL)                          592
PU (PPM > USL)                          1,852
P* (Total PPM Out-of-Spec.)             2,445
P* - Lower Confidence Limit             469
P* - Upper Confidence Limit             12,706
------------------------------         ------------
Cpa (Capability Index)                  1.010048
Cpa - Lower Confidence Limit            0.830635
Cpa - Upper Confidence Limit            1.165899
------------------------------         ------------
```

5.3.2 Bivariate Output File Example

A general bivariate output file example is shown below as Table 5.5.

Table 5.5
Bivariate Output File Example

UNIVARIATE STATISTICS	VARIABLE # 1	VARIABLE # 2
Selected Johnson Distribution	N	N
NS (Number of Samples)	100	100
LSL (Lower Spec. Limit)	-3.000000	-3.000000
USL (Upper Spec. Limit)	+3.000000	+3.000000
E (Xi)	0.000000	0.000000
L (Lambda)	+1.000000	+1.000000
D (Delta)	+1.029122	+0.970916
G (Gamma)	-0.094743	-0.076246
F-Value of K-S Test	57.05%	80.95%
PL (PPM < LSL)	731	1,400
PU (PPM > USL)	1,383	2,281
P* (Total PPM Out-of-Spec.)	2,114	3,680
P* - Lower Confidence Limit	464	940
P* - Upper Confidence Limit	10,611	15,485
Cpa (Capability Index)	1.024552	0.968115
Cpa - Lower Confidence Limit	0.851748	0.806940
Cpa - Upper Confidence Limit	1.166870	1.102647

MULTIVARIATE STATISTICS	SYSTEM	CORRELATION COEFFICIENT P12
Multivariate Johnson System	NN	+0.127664
MP* (Total PPM Out-of-Spec.)	5,806	
MP* - Lower Confidence Limit	2,367	
MP* - Upper Confidence Limit	15,887	
MCpa (Capability Index)	0.919511	
MCpa - Lower Confidence Limit	0.803836	
MCpa - Upper Confidence Limit	1.013298	

5.3.3 Trivariate Output File Example

A general trivariate output file example is shown as Table 5.6.

5.3.4 Quadrivariate Output File Example

A general quadrivariate output file example is shown as Table 5.7.

Table 5.6
Trivariate Output File Example

UNIVARIATE STATISTICS	VARIABLE # 1	VARIABLE # 2	VARIABLE # 3
Selected Johnson Distribution	N	N	N
NS (Number of Samples)	100	100	100
LSL (Lower Spec. Limit)	-3.000000	-3.000000	-3.000000
USL (Upper Spec. Limit)	+3.000000	+3.000000	+3.000000
E (Xi)	0.000000	0.000000	0.000000
L (Lambda)	+1.000000	+1.000000	+1.000000
D (Delta)	+0.857599	+1.044838	+1.018252
G (Gamma)	+0.061805	-0.070206	+0.096792
F-Value of K-S Test	95.62%	99.31%	99.13%
PL (PPM < LSL)	6,020	676	1,548
PU (PPM > USL)	4,212	1,091	812
P* (Total PPM Out-of-Spec.)	10,232	1,767	2,361
P* - Lower Confidence Limit	3,448	369	534
P* - Upper Confidence Limit	31,983	7,437	11,511
Cpa (Capability Index)	0.855968	1.042305	1.013541
Cpa - Lower Confidence Limit	0.714872	0.869913	0.842265
Cpa - Upper Confidence Limit	0.974906	1.187048	1.154452

MULTIVARIATE STATISTICS	SYSTEM	CORRELATION	COEFFICIENTS
Multivariate Johnson System	NNN	P12	-0.048276
MP* (Total PPM Out-of-Spec.)	14,489	P13	+0.037212
MP* - Lower Confidence Limit	6,759	P23	-0.095239
MP* - Upper Confidence Limit	33,732		
MCpa (Capability Index)	0.814968		
MCpa - Lower Confidence Limit	0.707754		
MCpa - Upper Confidence Limit	0.902825		

Table 5.7
Quadrivariate Output File Example

UNIVARIATE STATISTICS	VARIABLE # 1	VARIABLE # 2	VARIABLE # 3	VARIABLE # 4
Selected Johnson Distribution	N	N	N	N
NS (Number of Samples)	100	100	100	100
LSL (Lower Spec. Limit)	-3.000000	-3.000000	-3.000000	-3.000000
USL (Upper Spec. Limit)	+3.000000	+3.000000	+3.000000	+3.000000
E (Xi)	0.000000	0.000000	0.000000	0.000000
L (Lambda)	+1.000000	+1.000000	+1.000000	+1.000000
D (Delta)	+1.039267	+1.028953	+1.059260	+0.986340
G (Gamma)	-0.111229	+0.031615	+0.080422	+0.014291
F-Value of K-S Test	99.19%	94.34%	97.82%	84.16%
PL (PPM < LSL)	621	1,124	976	1,616
PU (PPM > USL)	1,321	909	561	1,473
P* (Total PPM Out-of-Spec.)	1,942	2,033	1,537	3,089
P* - Lower Confidence Limit	404	599	467	966
P* - Upper Confidence Limit	10,172	8,750	6,591	12,216
Cpa (Capability Index)	1.032992	1.028432	1.055887	0.986232
Cpa - Lower Confidence Limit	0.856647	0.863899	0.879882	0.830914
Cpa - Upper Confidence Limit	1.179287	1.171332	1.202570	1.123214

MULTIVARIATE STATISTICS	SYSTEM	CORRELATION	COEFFICIENTS
Multivariate Johnson System	NNNN	P12	-0.010485
MP* (Total PPM Out-of-Spec.)	8,512	P13	-0.075865
MP* - Lower Confidence Limit	3,323	P23	-0.025349
MP* - Upper Confidence Limit	24,005	P14	-0.059953
		P24	-0.083889
MCpa (Capability Index)	0.877020	P34	+0.057413
MCpa - Lower Confidence Limit	0.752346		
MCpa - Upper Confidence Limit	0.978713		

CHAPTER VI

CONCLUSION

6.1 Research Justification

The need to accurately describe a process's ability to create product within known specification limits is vital in modern industry. Process performance, commonly called process capability, drives many organizations in their goal to be profitable market leaders. Unfortunately, most of the tools available to industry today are severely limited in their abilities to estimate process capability. These limitations can be separated into two categories: (1) the process capability can only be estimated on one characteristic at a time, and (2) each process characteristic which is analyzed must be normally distributed. However, most processes are comprised of many characteristics, and correlation can exist between these characteristics when they are created by the same process. It is also the author's experience that there are many processes found in industry that have characteristics which are heavily skewed and/or have physical boundaries, and thus are not normally distributed. This research addresses the identified limitations on estimating process capability. This is done by estimating the capability of a univariate or multivariate process, without requirements placed on either the distribution's shape or correlation between variables.

6.2 Methodology Employed

The methodology employed in this research to obtain an accurate estimate of multivariate non-normal process capability is developed as follows:

1) Sample process data are analyzed and each marginal distribution is optimally fitted to one of five presented Johnson transformation systems, based on its descriptive statistics and a system selection decision matrix.
2) If the sample is multivariate, then the correlation between the transformed Johnson marginal distributions is calculated.
3) If the sample is univariate, then the proportion of nonconforming product is calculated. If the sample is multivariate, then the proportion of nonconforming product is calculated for each marginal distribution and the total proportion of nonconforming product for the multivariate process is estimated using multivariate simulation techniques. This process begins with generating a multivariate standard normal distribution with correlation as calculated in step 2. Each marginal distribution is reverse-transformed back into its original distribution using the calculated parameters from the selected Johnson systems. A count of the number of simulated samples which are considered nonconforming is transformed into a proportion nonconforming.
4) Confidence intervals on the univariate proportion or marginal and total proportions of nonconforming product are estimated.
5) The proportions of nonconforming product and their appropriate confidence interval estimates are transformed into a univariate capability index, C_{pn}, for univariate or

marginal distributions, and a multivariate capability index, MC_{pn}, for multivariate distributions.

6.3 Results and Insights

This research shows that a multivariate process will have a total proportion nonconforming less than or equal to the sum of the proportions nonconforming for each marginal distribution. It is also shown that correlation between marginal distributions actually increases the capability of a multivariate process. This increase was shown to be more pronounced for less capable processes. Performance of the multivariate correlated normal sample generator, the system selection decision matrix, and both process capability indices is demonstrated using selected sample cases with known results to compare with. The performance is also compared with results found with another approach from literature.

Changing the order of the input sample variables was shown to have no affect on the results. As computer processing speeds increase, the number of generated samples could also be dramatically increased. This increase in sample quantity would further reduce any error due to estimating a proportion by counting a finite sample.

6.4 Contributions

This research contributes to the current body of knowledge on process capability in several ways:

1) This research provides an estimate of the proportion of product which is being created as nonconforming to its specifications. This estimate is not limited to univariate processes nor normally distributed processes, and correlation is considered in the estimate, if it exists.
2) This research provides a multivariate capability index, MC_{pn}, which is related to the proportion of nonconforming product. This relationship allows a similar physical interpretation as with the widely-used univariate process capability indices; C_p and C_{pk}.
3) A software application is provided with this research that is both user-friendly and easy to use. This software application will allow process engineers and potential researchers to evaluate both the performance of their multivariate non-normal processes and the approach presented by this research.
4) This research provides a starting point for future advanced study on this important topic.

6.5 Future Research

Possible future advanced study can include any or all of the following topics:

1) Increase the level of performance of the current methodology through enhancements of the system selection decision matrix and the critical values of the decision variables, different goodness-of-fit tests, more powerful tests for normality.
2) Increase confidence interval performance through different alpha levels, system dependent alpha levels, or a different approach.

3) Include the ability to have dependent specifications or combinations of independent and dependent specifications.
4) Allow for more than four variables.
5) Create a different or a larger selection of transformation systems that can more adequately fit all distributions which are encountered in industry.

6.6 Summary

In conclusion, while it was the attempt of this research to allow industry personnel to begin understanding the actual performance of complex multivariate processes, it is probably more realistic that the non-normal applications of this research will find more use. I believe this probable result is due, in part, to difficulty in understanding multivariate statistics. I also see some difficulty in convincing top management that a complex multivariate process which has demonstrated capability on all or most of its individual key characteristics, can still be determined to not meet statistical capability requirements, based on its multivariate performance. There is, however, only one way to begin: Take the first step and try.

REFERENCES

Box, G. E. P. and Muller, M. E. (1958), "A Note on the Generation of Random Normal Deviates," *The Annals of Mathematical Statistics*, 29, 610-611.

Boyles, R. A. (1991a), "The Taguchi Capability Index," *Journal of Quality Technology*, 23, 17-26.

Boyles, R. A. (1991b), "Corrigenda," *Journal of Quality Technology*, 23, 382-383.

Chan, L. K., Cheng, S. W. and Spiring, F. A. (1988a), "A New Measure of Process Capability: C_{pm}," *Journal of Quality Technology*, 20, 162-175.

Chan, L. K., Cheng, S. W. and Spiring, F. A. (1988b), "A Graphical Technique Process Capability," *ASQC Quality Congress Transactions*, Dallas, 268-275.

Chan, L. K., Cheng, S. W. and Spiring, F. A. (1991), "A Multivariate Measure of Process Capability," *International Journal of Modeling and Simulation*, 11, 1-6. [Abridged Version in *Advances in Reliability and Quality Control* (1988) (M. Hamza ed.) ACTA Press, Anaheim, CA, 195-199].

Chen, H. (1994), "A Multivariate Process Capability Index Over a Rectangular Solid Tolerance Zone," *Statistica Sinica*, 4, 749-758.

Clements, J. A. (1989), "Process Capability Calculations for Non-Normal Distributions," *Quality Progress*, 9, 95-100.

Ebeling, C. E. (1997), *An Introduction to Reliability and Maintainability Engineering*, McGraw Hill.

Elderton, W. P. and Johnson, N. L. (1969), *Systems of Frequency Curves*, Cambridge University Press.

Hubele, N. F., Shahriari, H. and Cheng, C.S. (1991), "*A Bivariate Process Capability Vector*," in *Statistical Process Control in Manufacturing*, (J. B. Keats and D. C. Montgomery eds.), Marcel Dekker, NY, 299-310.

Johnson, M. E. (1987), *Multivariate Statistical Simulation*, John Wiley & Sons, Inc.

Johnson, N. L. and Kotz, S. (1970), *Distributions in Statistics: Continuous Univariate Distributions-1*, (H. Chernoff ed.), Houghton Mifflin Company, Boston, MA.

Johnson, N. L. and Kotz, S. (1972), *Distributions in Statistics: Continuous Multivariate Distributions*, John Wiley & Sons, Inc., NY.

Johnson, R. A. and Wichern, D. W. (1992), *Applied Multivariate Statistical Analysis*, 3rd edition, Prentice Hall, NJ.

Kane, V. E. (1986a), "Process Capability Indices," *Journal of Quality Technology*, 18, 41-52.

Kane, V. E. (1986b), "Corrigenda," *Journal of Quality Technology*, 18, 265.

Kocherlakota, S. and Kocherlakota, K. (1991), "Process Capability Index: Bivariate Normal Distribution," *Commun. Statist. Theory Methods*, 20, 2529-2547.

Kotz, S. and Johnson, N. L. (1993), *Process Capability Indices*, Chapman & Hall, NY, 179-201.

Littig, S. J., Lam, C. T. and Pollock, S. M. (1992), Capability Measurements for Multivariate Processes: Definitions and an Example for a Gear Carrier, *Technical Report* 92-42, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

Miller, D. W. (1995a), *Fitting Frequency Distributions: Philosophy and Practice – Part I: Discrete Distributions*, Book Resource.

Miller, D. W. (1995b), *Fitting Frequency Distributions: Philosophy and Practice – Part II: Continuous Distributions*, Book Resource.

Nelder, J. A. and Mead, R. (1965), "A Simplex Method for Function Minimization," *Computer Journal*, 7, 308-313.

O'Neill, R. (1971), "Algorithm AS 47: Function Minimization using a Simplex Procedure," *Applied Statistics*, 20, 338-345.

Pearn, W. L., Kotz, S. and Johnson, N. L. (1992), "Distribution and Inferential Properties of Process Capability Indices," *Journal of Quality Technology*, 24, 216-231.

Pomeranz, J. (1974), "Algorithm 487: Exact Cumulative Distribution of the Komogorov-Smirnov Statistic for Small Samples," *CACM*, 17, 703-704.

Rivera, L. A. R., Hubele, N. F. and Lawrence, F. P. (1995), "Cpk Index Estimation using Data Transformation," *Computers and Industrial Engineering*, 29, 55-58.

Siegel (1956), *Nonparametric Statistics for the Behavioral Sciences*, McGraw-Hill, NY.

Sullivan, L. P. (1984), "Reducing Variability: A New Approach to Quality," *Quality Progress*, 7, 15-21.

Sullivan, L. P. (1985), "Letters," *Quality Progress*, 4, 7-8.

Taam, W., Subbaiah, P. and Liddy, J. W. (1993), "A Note on Multivariate Capability Indices," *Journal of Applied Statistics*, 20, 339-351.

Taguchi, G. (1985), "A Tutorial on Quality Control and Assurance – the Taguchi Methods," *1985 ASA Annual Meeting*, Las Vegas, NV.

Tong, Y. L. (1990), *The Multivariate Normal Distribution*, Springer-Verlag, NY.

Wang, F. K., Hubele, N. F., Lawrence, F. P., Miskulin, J. D. and Shahriari, H. (1996), "Graphical Interpretations of Multivariate Process Capability," *5th Industrial Engineering Research Conference*, Minneapolis, MN.

Western Electric (1956), *Statistical Quality Control Handbook*, Western Electric Company, Indianapolis, IN.

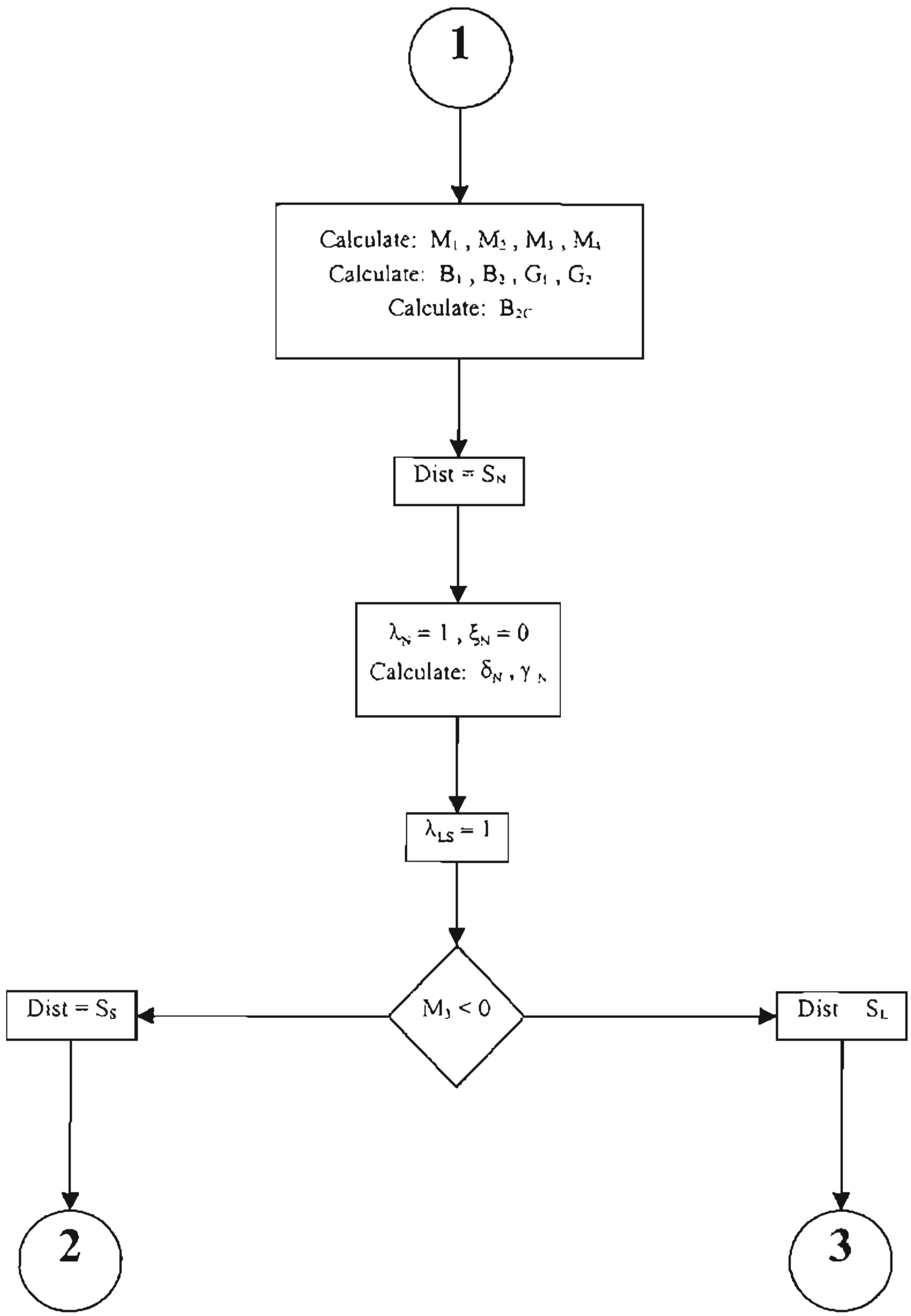
Wierda, S. J. (1993), "A Multivariate Process Capability Index," *ASQC Quality Congress Transactions*, Boston, 342-348.

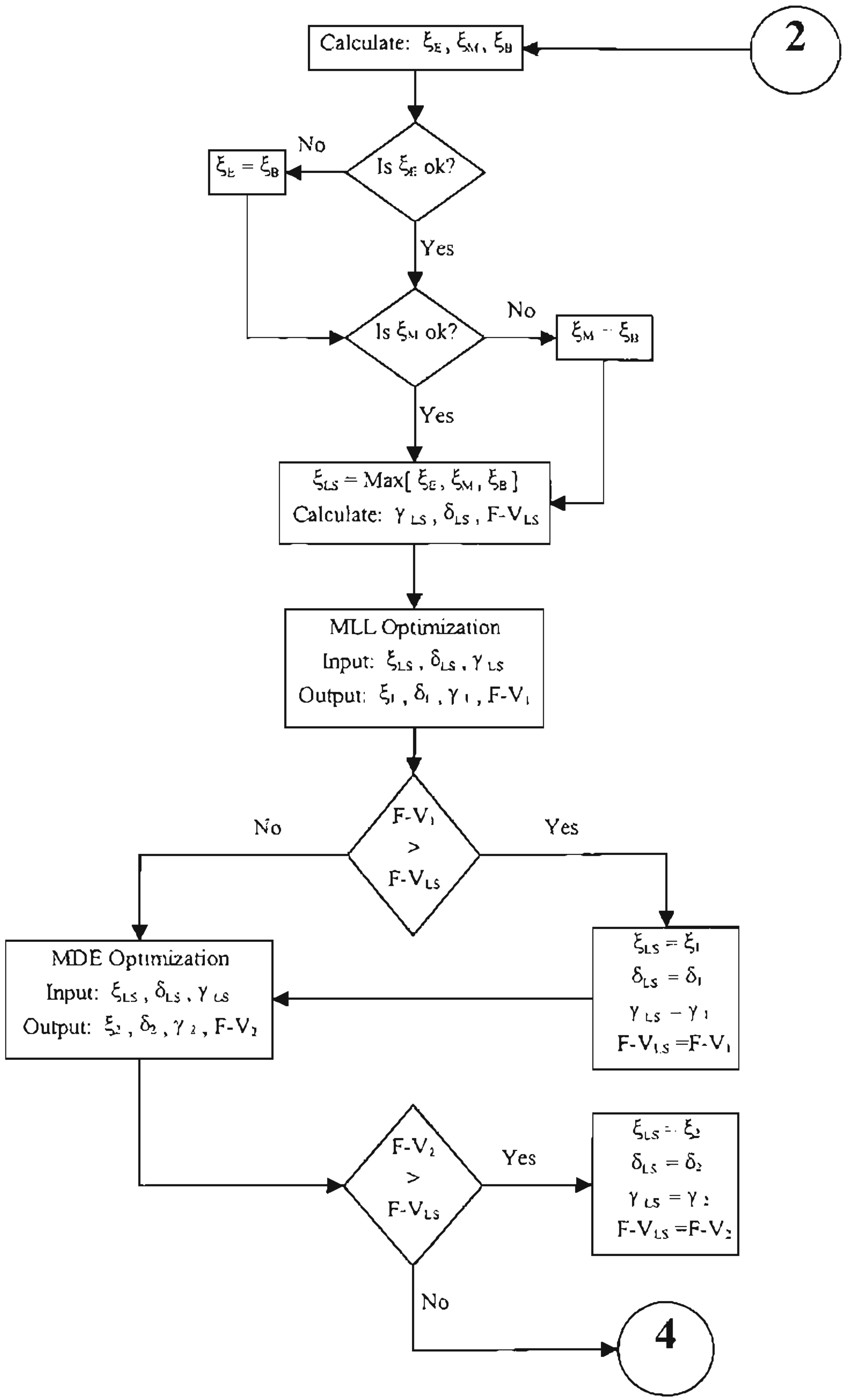
Wierda, S. J. (1994), "Multivariate Statistical Process Control: Recent Results and Directions for Future Research," *Statistica Neerlandica*, 48, 147-168.

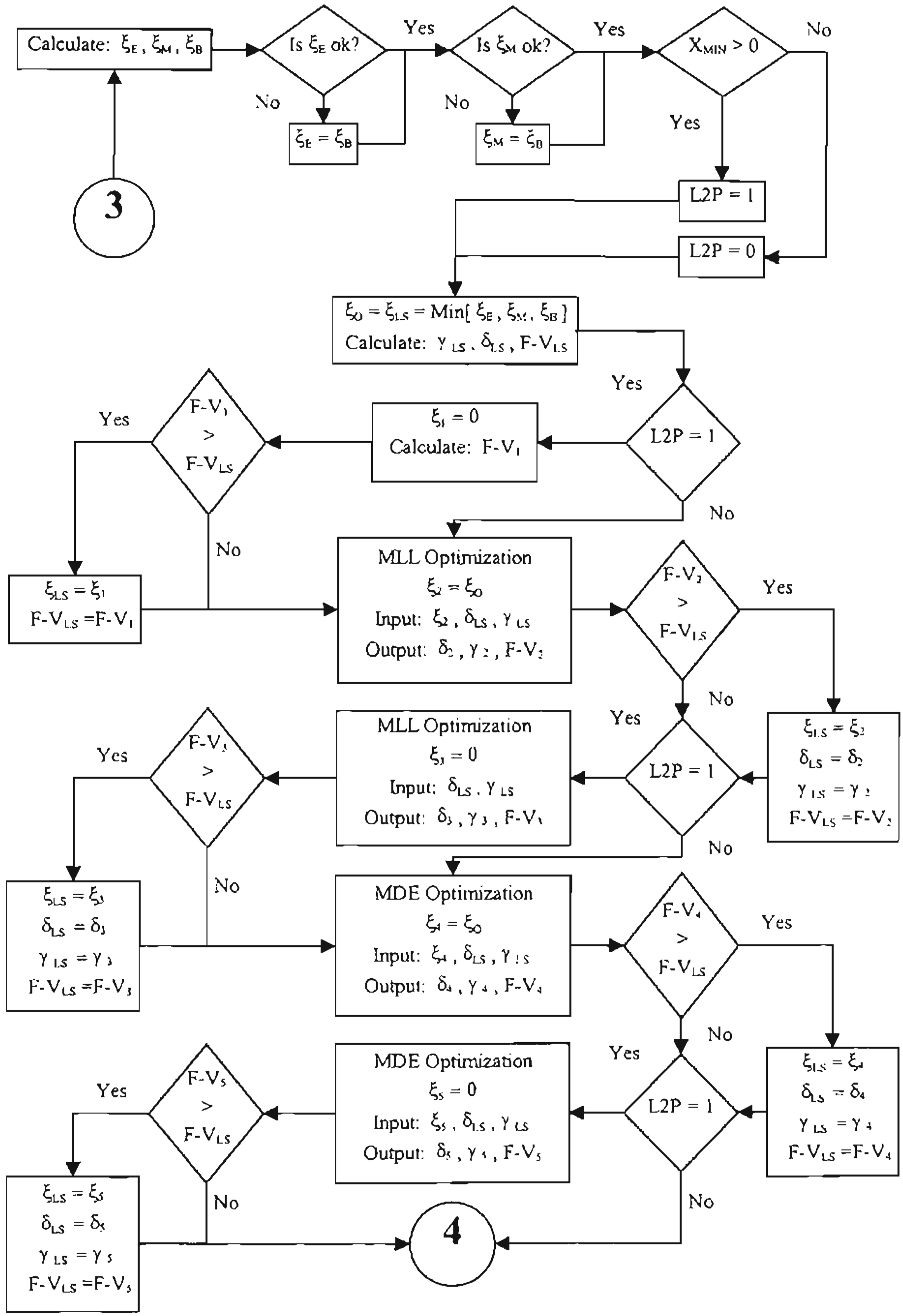
APPENDICES

APPENDIX A

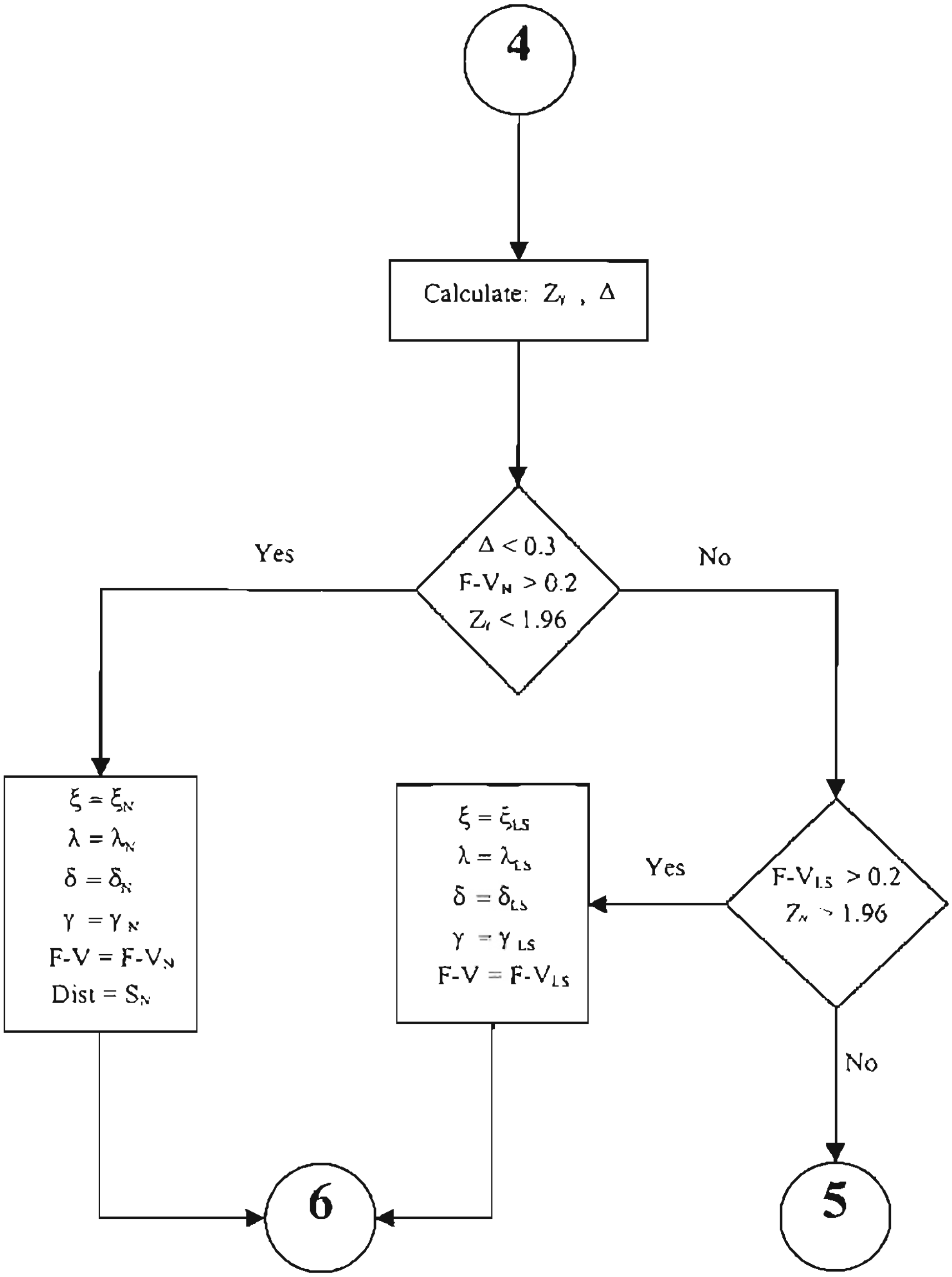
Flowcharts for Johnson System Fitting and Selection

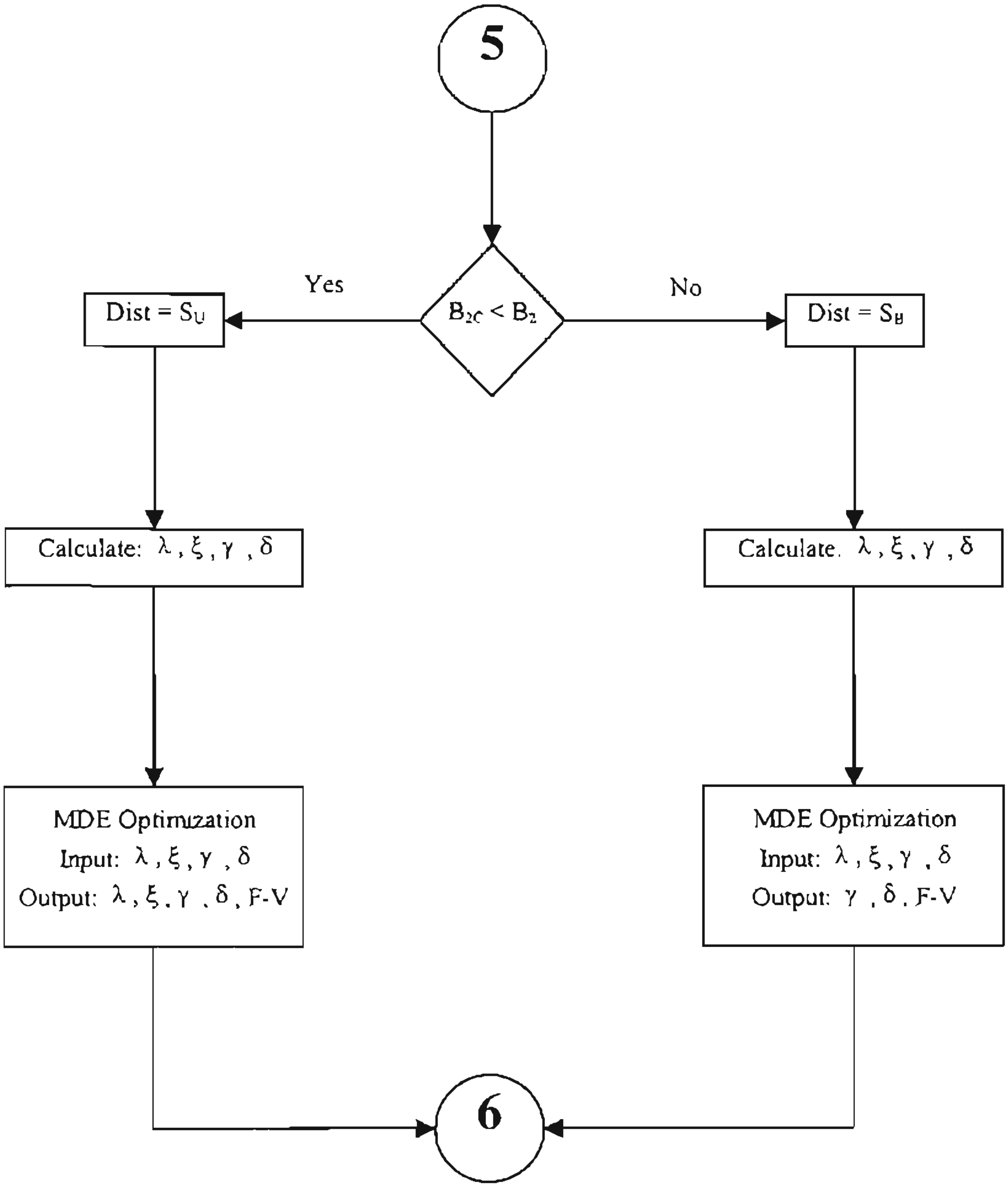

1
Calculate: M1 , M2 , M3 , M4
Calculate: B1 , B2 , G1 , G2
Calculate: B2C
Dist = SN
λN = 1 , ξN = 0
Calculate: δN , γN
λLS = 1
M3 < 0
Dist = SS
Dist SL
2
3


Calculate: ξ_E, ξ_M, ξ_B
2
Is ξ_E ok?
No
$\xi_E = \xi_B$
Yes
Is ξ_M ok?
No
$\xi_M = \xi_B$
Yes
ξ_{LS} = Max[ξ_E, ξ_M, ξ_B]
Calculate: γ_{LS}, δ_{LS}, F-V_{LS}
MLL Optimization
Input: ξ_{LS}, δ_{LS}, γ_{LS}
Output: ξ_1, δ_1, γ_1, F-V_1
F-V_1 > F-V_{LS}
No
Yes
$\xi_{LS} = \xi_1$
$\delta_{LS} = \delta_1$
$\gamma_{LS} = \gamma_1$
F-V_{LS} =F-V_1
MDE Optimization
Input: ξ_{LS}, δ_{LS}, γ_{LS}
Output: ξ_2, δ_2, γ_2, F-V_2
F-V_2 > F-V_{LS}
Yes
$\xi_{LS} = \xi_2$
$\delta_{LS} = \delta_2$
$\gamma_{LS} = \gamma_2$
F-V_{LS} =F-V_2
No
4



Calculate: ξ_E, ξ_M, ξ_B
Is ξ_E ok?
Yes
No
$\xi_E = \xi_B$
Is ξ_M ok?
Yes
No
$\xi_M = \xi_B$
$X_{MIN} > 0$
No
Yes
L2P = 1
L2P = 0
3
$\xi_O = \xi_{LS} = \mathrm{Min}[\xi_E, \xi_M, \xi_B]$
Calculate: γ_{LS}, δ_{LS}, F-V_{LS}
Yes
L2P = 1
No
$\xi_1 = 0$
Calculate: F-V_1
F-V_1 > F-V_{LS}
Yes
No
$\xi_{LS} = \xi_1$
F-V_{LS} =F-V_1
MLL Optimization
$\xi_2 = \xi_O$
Input: ξ_2, δ_{LS}, γ_{LS}
Output: δ_2, γ_2, F-V_2
F-V_2 > F-V_{LS}
Yes
No
$\xi_{LS} = \xi_2$
$\delta_{LS} = \delta_2$
$\gamma_{LS} = \gamma_2$
F-V_{LS} =F-V_2
Yes
L2P = 1
No
MLL Optimization
$\xi_3 = 0$
Input: δ_{LS}, γ_{LS}
Output: δ_3, γ_3, F-V_3
F-V_3 > F-V_{LS}
Yes
No
$\xi_{LS} = \xi_3$
$\delta_{LS} = \delta_3$
$\gamma_{LS} = \gamma_3$
F-V_{LS} =F-V_3
MDE Optimization
$\xi_4 = \xi_O$
Input: ξ_4, δ_{LS}, γ_{LS}
Output: δ_4, γ_4, F-V_4
F-V_4 > F-V_{LS}
Yes
No
$\xi_{LS} = \xi_4$
$\delta_{LS} = \delta_4$
$\gamma_{LS} = \gamma_4$
F-V_{LS} =F-V_4
Yes
L2P = 1
No
MDE Optimization
$\xi_5 = 0$
Input: ξ_5, δ_{LS}, γ_{LS}
Output: δ_5, γ_5, F-V_5
F-V_5 > F-V_{LS}
Yes
No
$\xi_{LS} = \xi_5$
$\delta_{LS} = \delta_5$
$\gamma_{LS} = \gamma_5$
F-V_{LS} =F-V_5
4


4
Calculate: Z_γ , Δ
$\Delta < 0.3$
$F\text{-}V_N > 0.2$
$Z_\gamma < 1.96$
Yes
No
$\xi = \xi_N$
$\lambda = \lambda_N$
$\delta = \delta_N$
$\gamma = \gamma_N$
$F\text{-}V = F\text{-}V_N$
Dist = S_N
$\xi = \xi_{LS}$
$\lambda = \lambda_{LS}$
$\delta = \delta_{LS}$
$\gamma = \gamma_{LS}$
$F\text{-}V = F\text{-}V_{LS}$
Yes
$F\text{-}V_{LS} > 0.2$
$Z_N > 1.96$
No
6
5


5
Yes
No
$B_{2C} < B_2$
Dist = S_U
Dist = S_B
Calculate: λ, ξ, γ, δ
Calculate: λ, ξ, γ, δ
MDE Optimization
Input: λ, ξ, γ, δ
Output: λ, ξ, γ, δ, F-V
MDE Optimization
Input: λ, ξ, γ, δ
Output: γ, δ, F-V
6

APPENDIX B

Description of Program Variables

Variable	Description
A21	Calculation variable in Z21, EZ21, and VZ21
A312	Calculation variable in Z312, EZ312, and VZ312
A4123	Calculation variable in Z4123, EZ4123, and VZ4123
ALPHA	Type 1 error value
B312	Calculation variable in Z312, EZ312, and VZ312
B4123	Calculation variable in Z4123, EZ4123, and VZ4123
C4123	Calculation variable in Z4123, EZ4123, and VZ4123
CHIH	Critical upper chi-square statistic
CHIL	Critical lower chi-square statistic
CNT(I,J,K,L)	Counter array for samples with respect to their specification limits: The first variable is represented as variable L and the last by variable I. If the index = 0 then the variable is not considered. If the index = 1 then the count is for under the lower specification limit. If the index = 2 then the count is for above the upper specification limit. If the index = 3 then the count is for within specification limits.
CPA(I,K)	Univariate process capability index of variable I at conf. rectangle point K
CPTOT(K1)	Confidence interval for univatiate process capability index
D(I)	Delta (δ) statistic for each variable
DIST(I)	Johnson system identifier for variable I
DP(I,K1)	Confidence interval and point estimate on delta statistic of variable I
DS(I,K)	Delta statistic of variable I at conf. rectangle point K
E(I)	Xi (ξ) statistic for each variable
EZ1	Independent expected value of Z1
EZ21	Conditional expected value of Z2 given Z1
EZ312	Conditional expected value of Z3 given Z1 and Z2
EZ4123	Conditional expected value of Z4 given Z1, Z2, and Z3
F4	Formatting variable for 4 decimal places
F6	Formatting variable for 6 decimal places
FV(I)	F-Value for each variable's fit
G(I)	Gamma (γ) statistic for each variable
GP(I,K1)	Confidence interval and point estimate on gamma statistic of variable I
GS(I,K)	Gamma statistic of variable I at conf. rectangle point K
I	Iteration variable, primarily for variable #
Ifilepath	Input file path (user-defined)
J	Iteration variable, primarily for sample #
K	Iteration variable, primarily for decision matrix position
K1	Iteration variable, primarily for confidence interval position
L(I)	Lambda (λ) statistic for each variable
LSL(I)	Lower-specification-limit for each variable
MAXPOS(I)	Conf. rectangle point representing upper confidence level of variable I
MCPA(K1)	Confidence interval for multivariate process capability index
MINPOS(I)	Conf. rectangle point representing lower confidence level of variable I
MPS	Total multivariate proportion nonconforming
NS	Number of samples
NV	Number of variables (1-4)

Variable	Description
Ofilepath	Output file pate (user-defined)
Openfile1	Available file number for input file
Openfile5	Available file number for output file
P12	Correlation coefficient between transformed variables 1 and 2
P13	Correlation coefficient between transformed variables 1 and 3
P14	Correlation coefficient between transformed variables 1 and 4
P23	Correlation coefficient between transformed variables 2 and 3
P24	Correlation coefficient between transformed variables 2 and 4
P34	Correlation coefficient between transformed variables 3 and 4
PL(I,K)	Lower proportion nonconforming of variable I at conf. rectangle point K
PS(I,K)	Total proportion nonconforming of variable I at conf. rectangle point K
PU(I,K)	Upper proportion nonconforming of variable I at conf. rectangle point K
S(I)	Calculated Johnson variate for each variable
SIM	Iteration variable, for simulation iterations
TLOW	Critical lower t statistic
USL(I)	Upper-specification-limit for each variable
VZ1	Independent variance of Z1
VZ21	Conditional variance of Z2 given Z1
VZ312	Conditional variance of Z3 given Z1 and Z2
VZ4123	Conditional variance of Z4 given Z1, Z2, and Z3
X1(J)	Sample values for variable 1
X2(J)	Sample values for variable 2, if NV > 1
X3(J)	Sample values for variable 3, if NV > 2
X4(J)	Sample values for variable 4, if NV = 4
Z(I,K)	Transformation of variable I at conf. rectangle point K
Z1	Independent multivariate standard normal variate Z1
Z21	Conditional multivariate standard normal variate Z2 given Z1
Z312	Conditional multivariate standard normal variate Z3 given Z1 and Z2
Z4123	Conditional multivariate standard normal variate Z4 given Z1, Z2, and Z3

APPENDIX C

Multivariate Process Capability Program Code

```
Sub MAIN()
'*************************************************************************
'* This Visual Basic Code Calculates the Univariate and Multivariate Process Capability
'* Indices, Cpa & MCpa, for a sample data set which is read.
'*************************************************************************
Dim NV As Integer, NS As Integer, I As Integer, J As Integer, K As Integer
Dim D() As Double, E() As Double, G() As Double, L() As Double
Dim S() As Double, FV() As Double, LSL() As Double, USL() As Double
Dim Z1 As Double, Z21 As Double, Z312 As Double, Z4123 As Double
Dim EZ1 As Double, EZ21 As Double, EZ312 As Double, EZ4123 As Double
Dim VZ1 As Double, VZ21 As Double, VZ312 As Double, VZ4123 As Double
Dim A21 As Double, A312 As Double, B312 As Double, A4123 As Double
Dim B4123 As Double, C4123 As Double, P12 As Double, P24 As Double
Dim P34 As Double, P13 As Double, P23 As Double, P14 As Double
Dim CHIL As Double, CHIH As Double, TLOW As Double, ALPHA As Double
Dim X1() As Double, X2() As Double, X3() As Double, X4() As Double
Dim PL() As Double, PU() As Double, PS() As Double, DIST() As String
Dim GS() As Double, DS() As Double, Z() As Double, CPA() As Double
Dim DP() As Double, GP() As Double, MINPOS() As Integer, MAXPOS() As Integer
Dim MCPA(1 To 9) As Double, CPTOT(1 To 9) As Double, MPS(1 To 3) As Long
Dim F4 As String, F6 As String, K1 As Integer, SIM As Long
Dim Openfile1 As Integer, Openfile5 As Integer, CNT(3, 3, 3, 3) As Long
Dim MySet As Recordset, MyDB As Database, Ifilepath As String, Ofilepath As String
F4 = "0.0000": F6 = "0.000000"
Const PI As Double = 3.14159265358979
DoCmd.Hourglass True
Set MyDB = CurrentDb
Set MySet = MyDB.OpenRecordset("File Specs", dbOpenTable)
Ifilepath = MySet![Input Totalpath]
Ofilepath = MySet![Output Totalpath]
MySet.Close

'************************* FIT SAMPLE POINTS *************************

Openfile5 = FreeFile
Open Ifilepath For Input As #Openfile5
Input #Openfile5, NV, NS
'Re-Dimensioning Dynamic Arrays for Memory Management
ReDim LSL(1 To NV): ReDim USL(1 To NV)
ReDim FV(1 To NV): ReDim S(1 To NV)
ReDim D(1 To NV): ReDim G(1 To NV)
ReDim L(1 To NV): ReDim E(1 To NV)
ReDim PL(1 To NV, 1 To 9): ReDim PU(1 To NV, 1 To 9)
ReDim PS(1 To NV, 1 To 9): ReDim DIST(1 To NV)
ReDim GS(1 To NV, 1 To 9): ReDim DS(1 To NV, 1 To 9)
ReDim Z(1 To NV, 1 To 9): ReDim CPA(1 To NV, 1 To 9)
```

```
ReDim DP(1 To NV, 1 To 3): ReDim GP(1 To NV, 1 To 3)
ReDim MINPOS(1 To NV): ReDim MAXPOS(1 To NV)
If NV = 1 Then
      ReDim X1(1 To NS): ReDim X2(0)
      ReDim X3(0): ReDim X4(0)
End If
If NV = 2 Then
      ReDim X1(1 To NS): ReDim X2(1 To NS)
      ReDim X3(0): ReDim X4(0)
End If
If NV = 3 Then
      ReDim X1(1 To NS): ReDim X2(1 To NS)
      ReDim X3(1 To NS): ReDim X4(0)
End If
If NV = 4 Then
      ReDim X1(1 To NS): ReDim X2(1 To NS)
      ReDim X3(1 To NS): ReDim X4(1 To NS)
End If

If NV = 1 Then Input #Openfile5, LSL(1), USL(1)
If NV = 2 Then Input #Openfile5, LSL(1), USL(1), LSL(2), USL(2)
If NV = 3 Then Input #Openfile5, LSL(1), USL(1), LSL(2), USL(2), _
                                    LSL(3), USL(3)
If NV = 4 Then Input #Openfile5, LSL(1), USL(1), LSL(2), USL(2), _
                                    LSL(3), USL(3), LSL(4), USL(4)

15 For I = 1 To NS
   If NV = 1 Then Input #Openfile5, X1(I)
   If NV = 2 Then Input #Openfile5, X1(I), X2(I)
   If NV = 3 Then Input #Openfile5, X1(I), X2(I), X3(I)
   If NV = 4 Then Input #Openfile5, X1(I), X2(I), X3(I), X4(I)
Next I
Close #Openfile5

Call DistFitter(X1(), NS, E(1), L(1), D(1), G(1), FV(1), DIST(1))
If NV = 1 Then GoTo 100
Call DistFitter(X2(), NS, E(2), L(2), D(2), G(2), FV(2), DIST(2))
P12 = CORRELXY(NS, DIST(1), X1(), G(1), D(1), E(1), L(1), _
                          DIST(2), X2(), G(2), D(2), E(2), L(2))
If NV = 2 Then GoTo 100
Call DistFitter(X3(), NS, E(3), L(3), D(3), G(3), FV(3), DIST(3))
P13 = CORRELXY(NS, DIST(1), X1(), G(1), D(1), E(1), L(1), _
                          DIST(3), X3(), G(3), D(3), E(3), L(3))
P23 = CORRELXY(NS, DIST(2), X2(), G(2), D(2), E(2), L(2), _
                          DIST(3), X3(), G(3), D(3), E(3), L(3))
If NV = 3 Then GoTo 100
```

```
Call DistFitter(X4(), NS, E(4), L(4), D(4), G(4), FV(4), DIST(4))
P14 = CORRELXY(NS, DIST(1), X1(), G(1), D(1), E(1), L(1), _
                              DIST(4), X4(), G(4), D(4), E(4), L(4))
P24 = CORRELXY(NS, DIST(2), X2(), G(2), D(2), E(2), L(2), _
                              DIST(4), X4(), G(4), D(4), E(4), L(4))
P34 = CORRELXY(NS, DIST(3), X3(), G(3), D(3), E(3), L(3), _
                              DIST(4), X4(), G(4), D(4), E(4), L(4))

100 '*************UNIVARIATE CONFIDENCE INTERVALS******************
ALPHA = 0.05
CHIL = Excel.Application.WorksheetFunction.ChiInv((ALPHA / 2), NS - 1)
CHIH = Excel.Application.WorksheetFunction.ChiInv((1 - ALPHA / 2), NS - 1)
TLOW = Excel.Application.WorksheetFunction.TInv(ALPHA, NS - 1)
For I = 1 To NV
DS(I, 4) = D(I)
DS(I, 1) = DS(I, 4) * Sqr(CHIH / (NS - 1))
DS(I, 7) = DS(I, 4) * Sqr(CHIL / (NS - 1))
DS(I, 2) = DS(I, 1): DS(I, 3) = DS(I, 1)
DS(I, 5) = DS(I, 7): DS(I, 6) = DS(I, 7)
GS(I, 4) = G(I)
GS(I, 1) = GS(I, 4) - (TLOW / Sqr(NS))
GS(I, 7) = GS(I, 4) + (TLOW / Sqr(NS))
GS(I, 2) = GS(I, 4): GS(I, 3) = GS(I, 7)
GS(I, 5) = GS(I, 1): GS(I, 6) = GS(I, 4)
For K = 1 To 7
Select Case DIST(I)

   Case "N" 'Normal
      Z(I, K) = GS(I, K) + DS(I, K) * LSL(I)
      PL(I, K) = Excel.Application.WorksheetFunction.NormSDist(Z(I, K))
      Z(I, K) = GS(I, K) + DS(I, K) * USL(I)
      PU(I, K) = 1 - Excel.Application.WorksheetFunction.NormSDist(Z(I, K))
      PS(I, K) = PL(I, K) + PU(I, K)
      CPA(I, K) = Excel.Application.WorksheetFunction.NormSInv(1 - PS(I, K) / 2) / 3

   Case "L" 'LogNormal
   If E(I) < LSL(I) Then
      Z(I, K) = GS(I, K) + DS(I, K) * Log(LSL(I) - E(I))
      PL(I, K) = Excel.Application.WorksheetFunction.NormSDist(Z(I, K))
   Else
      PL(I, K) = 0
   End If
   If E(I) < USL(I) Then
      Z(I, K) = GS(I, K) + DS(I, K) * Log(USL(I) - E(I))
      PU(I, K) = 1 - Excel.Application.WorksheetFunction.NormSDist(Z(I, K))
   Else
```

```
  PU(I, K) = 0
End If
  PS(I, K) = PL(I, K) + PU(I, K)
If PS(I, K) > 0 Then
  CPA(I, K) = Excel.Application.WorksheetFunction.NormSInv(1 - PS(I, K) / 2) / 3
Else
  CPA(I, K) = 2
End If

Case "S" 'Special
If E(I) > LSL(I) Then
  Z(I, K) = GS(I, K) + DS(I, K) * Log(E(I) - LSL(I))
  PL(I, K) = 1 - Excel.Application.WorksheetFunction.NormSDist(Z(I, K))
Else
  PL(I, K) = 0
End If
If E(I) > USL(I) Then
  Z(I, K) = GS(I, K) + DS(I, K) * Log(E(I) - USL(I))
  PU(I, K) = Excel.Application.WorksheetFunction.NormSDist(Z(I, K))
Else
  PU(I, K) = 0
End If
  PS(I, K) = PL(I, K) + PU(I, K)
If PS(I, K) > 0 Then
  CPA(I, K) = Excel.Application.WorksheetFunction.NormSInv(1 - PS(I, K) / 2) / 3
Else
  CPA(I, K) = 2
End If

Case "B" 'Bounded
If E(I) < LSL(I) And LSL(I) < E(I) + L(I) Then
  Z(I, K) = GS(I, K) + DS(I, K) * Log((LSL(I) - E(I)) / (E(I) + L(I) - LSL(I)))
  PL(I, K) = Excel.Application.WorksheetFunction.NormSDist(Z(I, K))
Else
  PL(I, K) = 0
End If
If E(I) < USL(I) And USL(I) < E(I) + L(I) Then
  Z(I, K) = GS(I, K) + DS(I, K) * Log((USL(I) - E(I)) / (E(I) + L(I) - USL(I)))
  PU(I, K) = 1 - Excel.Application.WorksheetFunction.NormSDist(Z(I, K))
Else
  PU(I, K) = 0
End If
  PS(I, K) = PL(I, K) + PU(I, K)
If PS(I, K) > 0 Then
  CPA(I, K) = Excel.Application.WorksheetFunction.NormSInv(1 - PS(I, K) / 2) / 3
Else
```

```
        CPA(I, K) = 2
    End If

    Case "U" 'UnBounded
        Z(I, K) = GS(I, K) + DS(I, K) * ArcSinh((LSL(I) - E(I)) / L(I))
        PL(I, K) = Excel.Application.WorksheetFunction.NormSDist(Z(I, K))
        Z(I, K) = GS(I, K) + DS(I, K) * ArcSinh((USL(I) - E(I)) / L(I))
        PU(I, K) = 1 - Excel.Application.WorksheetFunction.NormSDist(Z(I, K))
        PS(I, K) = PL(I, K) + PU(I, K)
        CPA(I, K) = Excel.Application.WorksheetFunction.NormSInv(1 - PS(I, K) / 2) / 3

End Select
Next K
MINPOS(I) = MIN3POS(CPA(I, 1), CPA(I, 2), CPA(I, 3))
MAXPOS(I) = MAX3POS(CPA(I, 5), CPA(I, 6), CPA(I, 7))
Next I

If NV = 1 Then GoTo 125 'IF UNIVARIATE, SKIP SIMULATION

'ADJUSTED FOR MULTIVARIATE SIMULTANEOUS CONFIDENCE INTERVALS
ALPHA = 0.2
CHIL = Excel.Application.WorksheetFunction.ChiInv((ALPHA / 2), NS - 1)
CHIH = Excel.Application.WorksheetFunction.ChiInv((1 - ALPHA / 2), NS - 1)
TLOW = Excel.Application.WorksheetFunction.TInv(ALPHA, NS - 1)
For I = 1 To NV
DS(I, 1) = DS(I, 4) * Sqr(CHIH / (NS - 1))
DS(I, 7) = DS(I, 4) * Sqr(CHIL / (NS - 1))
DS(I, 2) = DS(I, 1): DS(I, 3) = DS(I, 1)
DS(I, 5) = DS(I, 7): DS(I, 6) = DS(I, 7)
GS(I, 1) = GS(I, 4) - (TLOW / Sqr(NS))
GS(I, 7) = GS(I, 4) + (TLOW / Sqr(NS))
GS(I, 2) = GS(I, 4): GS(I, 3) = GS(I, 7)
GS(I, 5) = GS(I, 1): GS(I, 6) = GS(I, 4)
DP(I, 1) = DS(I, MINPOS(I))
DP(I, 2) = DS(I, 4)
DP(I, 3) = DS(I, MAXPOS(I))
GP(I, 1) = GS(I, MINPOS(I))
GP(I, 2) = GS(I, 4)
GP(I, 3) = GS(I, MAXPOS(I))
Next I

For K1 = 1 To 3

Erase CNT 'REINITIALIZES THE COUNTER ARRAY

For SIM = 1 To 1000000
```

```
EZ1 = 0
VZ1 = 1
Z1 = EZ1 + Sqr(VZ1) * Sqr(-2 * Log(Rnd())) * Cos(2 * PI * Rnd())
Select Case DIST(1)
   Case "N" 'Normal - (N***)
   S(1) = (Z1 - GP(1, K1)) / DP(1, K1)
   Case "L" 'LogNormal - (L***)
   S(1) = Exp((Z1 - GP(1, K1)) / DP(1, K1)) + E(1)
   Case "S" 'Special (S***)
   S(1) = E(1) - Exp((Z1 - GP(1, K1)) / DP(1, K1))
   Case "B" 'Bounded - (B***)
   S(1) = L(1) * (1 + Exp((GP(1, K1) - Z1) / DP(1, K1))) ^ (-1) + E(1)
   Case "U" 'UnBounded - (U***)
   S(1) = L(1) * Sinh((Z1 - GP(1, K1)) / DP(1, K1)) + E(1)
End Select
   If S(1) < LSL(1) Then CNT(0, 0, 0, 1) = CNT(0, 0, 0, 1) + 1
   If S(1) > USL(1) Then CNT(0, 0, 0, 2) = CNT(0, 0, 0, 2) + 1

   If NV = 1 Then GoTo 105

A21 = P12 / VZ1
EZ21 = (A21 * Z1)
VZ21 = 1 - (A21 * P12)
Z21 = EZ21 + Sqr(VZ21) * Sqr(-2 * Log(Rnd())) * Cos(2 * PI * Rnd())
Select Case DIST(2)
   Case "N" 'Normal - (*N**)
   S(2) = (Z21 - GP(2, K1)) / DP(2, K1)
   Case "L" 'LogNormal - (*L**)
   S(2) = Exp((Z21 - GP(2, K1)) / DP(2, K1)) + E(2)
   Case "S" 'Special - (*S**)
   S(2) = E(2) - Exp((Z21 - GP(2, K1)) / DP(2, K1))
   Case "B" 'Bounded - (*B**)
   S(2) = L(2) * (1 + Exp((GP(2, K1) - Z21) / DP(2, K1))) ^ (-1) + E(2)
   Case "U" 'UnBounded - (*U**)
   S(2) = L(2) * Sinh((Z21 - GP(2, K1)) / DP(2, K1)) + E(2)
End Select
   If S(2) < LSL(2) Then
      CNT(0, 0, 1, 0) = CNT(0, 0, 1, 0) + 1
      If S(1) < LSL(1) Then CNT(0, 0, 1, 1) = CNT(0, 0, 1, 1) + 1
      If S(1) > USL(1) Then CNT(0, 0, 1, 2) = CNT(0, 0, 1, 2) + 1
   ElseIf S(2) > USL(2) Then
      CNT(0, 0, 2, 0) = CNT(0, 0, 2, 0) + 1
      If S(1) < LSL(1) Then CNT(0, 0, 2, 1) = CNT(0, 0, 2, 1) + 1
      If S(1) > USL(1) Then CNT(0, 0, 2, 2) = CNT(0, 0, 2, 2) + 1
   Else
```

```
      CNT(0, 0, 3, 0) = CNT(0, 0, 3, 0) + 1
      If S(1) < LSL(1) Then CNT(0, 0, 3, 1) = CNT(0, 0, 3, 1) + 1
      If S(1) > USL(1) Then CNT(0, 0, 3, 2) = CNT(0, 0, 3, 2) + 1
   End If

   If NV = 2 Then GoTo 105

A312 = (P13 - P12 * P23) / (VZ21 * VZ1)
B312 = (P23 - P12 * P13) / (VZ21 * VZ1)
EZ312 = (A312 * Z1) + (B312 * Z21)
VZ312 = 1 - ((A312 * P13) + (B312 * P23))
Z312 = EZ312 + Sqr(VZ312) * Sqr(-2 * Log(Rnd())) * Cos(2 * PI * Rnd())
Select Case DIST(3)
   Case "N" 'Normal - (**N*)
   S(3) = (Z312 - GP(3, K1)) / DP(3, K1)
   Case "L" 'LogNormal - (**L*)
   S(3) = Exp((Z312 - GP(3, K1)) / DP(3, K1)) + E(3)
   Case "S" 'Special - (**S*)
   S(3) = E(3) - Exp((Z312 - GP(3, K1)) / DP(3, K1))
   Case "B" 'Bounded - (**B*)
   S(3) = L(3) * (1 + Exp((GP(3, K1) - Z312) / DP(3, K1))) ^ (-1) + E(3)
   Case "U" 'UnBounded - (**U*)
   S(3) = L(3) * Sinh((Z312 - GP(3, K1)) / DP(3, K1)) + E(3)
End Select
   If S(3) < LSL(3) Then
      CNT(0, 1, 0, 0) = CNT(0, 1, 0, 0) + 1
      If S(2) < LSL(2) Then
         CNT(0, 1, 1, 0) = CNT(0, 1, 1, 0) + 1
         If S(1) < LSL(1) Then CNT(0, 1, 1, 1) = CNT(0, 1, 1, 1) + 1
         If S(1) > USL(1) Then CNT(0, 1, 1, 2) = CNT(0, 1, 1, 2) + 1
      ElseIf S(2) > USL(2) Then
         CNT(0, 1, 2, 0) = CNT(0, 1, 2, 0) + 1
         If S(1) < LSL(1) Then CNT(0, 1, 2, 1) = CNT(0, 1, 2, 1) + 1
         If S(1) > USL(1) Then CNT(0, 1, 2, 2) = CNT(0, 1, 2, 2) + 1
      Else
         CNT(0, 1, 3, 0) = CNT(0, 1, 3, 0) + 1
         If S(1) < LSL(1) Then CNT(0, 1, 3, 1) = CNT(0, 1, 3, 1) + 1
         If S(1) > USL(1) Then CNT(0, 1, 3, 2) = CNT(0, 1, 3, 2) + 1
      End If
   ElseIf S(3) > USL(3) Then
      CNT(0, 2, 0, 0) = CNT(0, 2, 0, 0) + 1
      If S(2) < LSL(2) Then
         CNT(0, 2, 1, 0) = CNT(0, 2, 1, 0) + 1
         If S(1) < LSL(1) Then CNT(0, 2, 1, 1) = CNT(0, 2, 1, 1) + 1
         If S(1) > USL(1) Then CNT(0, 2, 1, 2) = CNT(0, 2, 1, 2) + 1
      ElseIf S(2) > USL(2) Then
```

```
            CNT(0, 2, 2, 0) = CNT(0, 2, 2, 0) + 1
            If S(1) < LSL(1) Then CNT(0, 2, 2, 1) = CNT(0, 2, 2, 1) + 1
            If S(1) > USL(1) Then CNT(0, 2, 2, 2) = CNT(0, 2, 2, 2) + 1
        Else
            CNT(0, 2, 3, 0) = CNT(0, 2, 3, 0) + 1
            If S(1) < LSL(1) Then CNT(0, 2, 3, 1) = CNT(0, 2, 3, 1) + 1
            If S(1) > USL(1) Then CNT(0, 2, 3, 2) = CNT(0, 2, 3, 2) + 1
        End If
    Else
        CNT(0, 3, 0, 0) = CNT(0, 3, 0, 0) + 1
        If S(2) < LSL(2) Then
            CNT(0, 3, 1, 0) = CNT(0, 3, 1, 0) + 1
            If S(1) < LSL(1) Then CNT(0, 3, 1, 1) = CNT(0, 3, 1, 1) + 1
            If S(1) > USL(1) Then CNT(0, 3, 1, 2) = CNT(0, 3, 1, 2) + 1
        ElseIf S(2) > USL(2) Then
            CNT(0, 3, 2, 0) = CNT(0, 3, 2, 0) + 1
            If S(1) < LSL(1) Then CNT(0, 3, 2, 1) = CNT(0, 3, 2, 1) + 1
            If S(1) > USL(1) Then CNT(0, 3, 2, 2) = CNT(0, 3, 2, 2) + 1
        Else
            CNT(0, 3, 3, 0) = CNT(0, 3, 3, 0) + 1
            If S(1) < LSL(1) Then CNT(0, 3, 3, 1) = CNT(0, 3, 3, 1) + 1
            If S(1) > USL(1) Then CNT(0, 3, 3, 2) = CNT(0, 3, 3, 2) + 1
        End If
    End If

    If NV = 3 Then GoTo 105

A4123 = (P14 * (1 - P23 ^ 2) + P24 * (P13 * P23 - P12) + P34 * (P12 * P23 - P13)) / _
        (VZ312 * VZ21 * VZ1)
B4123 = (P14 * (P13 * P23 - P12) + P24 * (1 - P13 ^ 2) + P34 * (P12 * P13 - P23)) / _
        (VZ312 * VZ21 * VZ1)
C4123 = (P14 * (P12 * P23 - P13) + P24 * (P12 * P13 - P23) + P34 * (1 - P12 ^ 2)) / _
        (VZ312 * VZ21 * VZ1)
EZ4123 = (A4123 * Z1) + (B4123 * Z21) + (C4123 * Z312)
VZ4123 = 1 - ((A4123 * P14) + (B4123 * P24) + (C4123 * P34))
Z4123 = EZ4123 + Sqr(VZ4123) * Sqr(-2 * Log(Rnd())) * Cos(2 * PI * Rnd())
Select Case DIST(4)
    Case "N" 'Normal - (***N)
    S(4) = (Z4123 - GP(4, K1)) / DP(4, K1)
    Case "L" 'LogNormal - (***L)
    S(4) = Exp((Z4123 - GP(4, K1)) / DP(4, K1)) + E(4)
    Case "S" 'Special - (***S)
    S(4) = E(4) - Exp((Z4123 - GP(4, K1)) / DP(4, K1))
    Case "B" 'Bounded - (***B)
    S(4) = L(4) * (1 + Exp((GP(4, K1) - Z4123) / DP(4, K1))) ^ (-1) + E(4)
    Case "U" 'UnBounded - (***U)
```

```
  S(4) = L(4) * Sinh((Z4123 - GP(4, K1)) / DP(4, K1)) + E(4)
End Select
  If S(4) < LSL(4) Then
    CNT(1, 0, 0, 0) = CNT(1, 0, 0, 0) + 1
    If S(3) < LSL(3) Then
      CNT(1, 1, 0, 0) = CNT(1, 1, 0, 0) + 1
      If S(2) < LSL(2) Then
        CNT(1, 1, 1, 0) = CNT(1, 1, 1, 0) + 1
        If S(1) < LSL(1) Then CNT(1, 1, 1, 1) = CNT(1, 1, 1, 1) + 1
        If S(1) > USL(1) Then CNT(1, 1, 1, 2) = CNT(1, 1, 1, 2) + 1
      ElseIf S(2) > USL(2) Then
        CNT(1, 1, 2, 0) = CNT(1, 1, 2, 0) + 1
        If S(1) < LSL(1) Then CNT(1, 1, 2, 1) = CNT(1, 1, 2, 1) + 1
        If S(1) > USL(1) Then CNT(1, 1, 2, 2) = CNT(1, 1, 2, 2) + 1
      Else
        CNT(1, 1, 3, 0) = CNT(1, 1, 3, 0) + 1
        If S(1) < LSL(1) Then CNT(1, 1, 3, 1) = CNT(1, 1, 3, 1) + 1
        If S(1) > USL(1) Then CNT(1, 1, 3, 2) = CNT(1, 1, 3, 2) + 1
      End If
    ElseIf S(3) > USL(3) Then
      CNT(1, 2, 0, 0) = CNT(1, 2, 0, 0) + 1
      If S(2) < LSL(2) Then
        CNT(1, 2, 1, 0) = CNT(1, 2, 1, 0) + 1
        If S(1) < LSL(1) Then CNT(1, 2, 1, 1) = CNT(1, 2, 1, 1) + 1
        If S(1) > USL(1) Then CNT(1, 2, 1, 2) = CNT(1, 2, 1, 2) + 1
      ElseIf S(2) > USL(2) Then
        CNT(1, 2, 2, 0) = CNT(1, 2, 2, 0) + 1
        If S(1) < LSL(1) Then CNT(1, 2, 2, 1) = CNT(1, 2, 2, 1) + 1
        If S(1) > USL(1) Then CNT(1, 2, 2, 2) = CNT(1, 2, 2, 2) + 1
      Else
        CNT(1, 2, 3, 0) = CNT(1, 2, 3, 0) + 1
        If S(1) < LSL(1) Then CNT(1, 2, 3, 1) = CNT(1, 2, 3, 1) + 1
        If S(1) > USL(1) Then CNT(1, 2, 3, 2) = CNT(1, 2, 3, 2) + 1
      End If
    Else
      CNT(1, 3, 0, 0) = CNT(1, 3, 0, 0) + 1
      If S(2) < LSL(2) Then
        CNT(1, 3, 1, 0) = CNT(1, 3, 1, 0) + 1
        If S(1) < LSL(1) Then CNT(1, 3, 1, 1) = CNT(1, 3, 1, 1) + 1
        If S(1) > USL(1) Then CNT(1, 3, 1, 2) = CNT(1, 3, 1, 2) + 1
      ElseIf S(2) > USL(2) Then
        CNT(1, 3, 2, 0) = CNT(1, 3, 2, 0) + 1
        If S(1) < LSL(1) Then CNT(1, 3, 2, 1) = CNT(1, 3, 2, 1) + 1
        If S(1) > USL(1) Then CNT(1, 3, 2, 2) = CNT(1, 3, 2. 2) + 1
      Else
        CNT(1, 3, 3, 0) = CNT(1, 3, 3, 0) + 1
```

```
        If S(1) < LSL(1) Then CNT(1, 3, 3, 1) = CNT(1, 3, 3, 1) + 1
        If S(1) > USL(1) Then CNT(1, 3, 3, 2) = CNT(1, 3, 3, 2) + 1
      End If
    End If
  ElseIf S(4) > USL(4) Then
    CNT(2, 0, 0, 0) = CNT(2, 0, 0, 0) + 1
    If S(3) < LSL(3) Then
      CNT(2, 1, 0, 0) = CNT(2, 1, 0, 0) + 1
      If S(2) < LSL(2) Then
        CNT(2, 1, 1, 0) = CNT(2, 1, 1, 0) + 1
        If S(1) < LSL(1) Then CNT(2, 1, 1, 1) = CNT(2, 1, 1, 1) + 1
        If S(1) > USL(1) Then CNT(2, 1, 1, 2) = CNT(2, 1, 1, 2) + 1
      ElseIf S(2) > USL(2) Then
        CNT(2, 1, 2, 0) = CNT(2, 1, 2, 0) + 1
        If S(1) < LSL(1) Then CNT(2, 1, 2, 1) = CNT(2, 1, 2, 1) + 1
        If S(1) > USL(1) Then CNT(2, 1, 2, 2) = CNT(2, 1, 2, 2) + 1
      Else
        CNT(2, 1, 3, 0) = CNT(2, 1, 3, 0) + 1
        If S(1) < LSL(1) Then CNT(2, 1, 3, 1) = CNT(2, 1, 3, 1) + 1
        If S(1) > USL(1) Then CNT(2, 1, 3, 2) = CNT(2, 1, 3, 2) + 1
      End If
    ElseIf S(3) > USL(3) Then
      CNT(2, 2, 0, 0) = CNT(2, 2, 0, 0) + 1
      If S(2) < LSL(2) Then
        CNT(2, 2, 1, 0) = CNT(2, 2, 1, 0) + 1
        If S(1) < LSL(1) Then CNT(2, 2, 1, 1) = CNT(2, 2, 1, 1) + 1
        If S(1) > USL(1) Then CNT(2, 2, 1, 2) = CNT(2, 2, 1, 2) + 1
      ElseIf S(2) > USL(2) Then
        CNT(2, 2, 2, 0) = CNT(2, 2, 2, 0) + 1
        If S(1) < LSL(1) Then CNT(2, 2, 2, 1) = CNT(2, 2, 2, 1) + 1
        If S(1) > USL(1) Then CNT(2, 2, 2, 2) = CNT(2, 2, 2, 2) + 1
      Else
        CNT(2, 2, 3, 0) = CNT(2, 2, 3, 0) + 1
        If S(1) < LSL(1) Then CNT(2, 2, 3, 1) = CNT(2, 2, 3, 1) + 1
        If S(1) > USL(1) Then CNT(2, 2, 3, 2) = CNT(2, 2, 3, 2) + 1
      End If
    Else
      CNT(2, 3, 0, 0) = CNT(2, 3, 0, 0) + 1
      If S(2) < LSL(2) Then
        CNT(2, 3, 1, 0) = CNT(2, 3, 1, 0) + 1
        If S(1) < LSL(1) Then CNT(2, 3, 1, 1) = CNT(2, 3, 1, 1) + 1
        If S(1) > USL(1) Then CNT(2, 3, 1, 2) = CNT(2, 3, 1, 2) + 1
      ElseIf S(2) > USL(2) Then
        CNT(2, 3, 2, 0) = CNT(2, 3, 2, 0) + 1
        If S(1) < LSL(1) Then CNT(2, 3, 2, 1) = CNT(2, 3, 2, 1) + 1
        If S(1) > USL(1) Then CNT(2, 3, 2, 2) = CNT(2, 3, 2, 2) + 1
```

```
        Else
          CNT(2, 3, 3, 0) = CNT(2, 3, 3, 0) + 1
          If S(1) < LSL(1) Then CNT(2, 3, 3, 1) = CNT(2, 3, 3, 1) + 1
          If S(1) > USL(1) Then CNT(2, 3, 3, 2) = CNT(2, 3, 3, 2) + 1
        End If
      End If
    Else
      CNT(3, 0, 0, 0) = CNT(3, 0, 0, 0) + 1
      If S(3) < LSL(3) Then
        CNT(3, 1, 0, 0) = CNT(3, 1, 0, 0) + 1
        If S(2) < LSL(2) Then
          CNT(3, 1, 1, 0) = CNT(3, 1, 1, 0) + 1
          If S(1) < LSL(1) Then CNT(3, 1, 1, 1) = CNT(3, 1, 1, 1) + 1
          If S(1) > USL(1) Then CNT(3, 1, 1, 2) = CNT(3, 1, 1, 2) + 1
        ElseIf S(2) > USL(2) Then
          CNT(3, 1, 2, 0) = CNT(3, 1, 2, 0) + 1
          If S(1) < LSL(1) Then CNT(3, 1, 2, 1) = CNT(3, 1, 2, 1) + 1
          If S(1) > USL(1) Then CNT(3, 1, 2, 2) = CNT(3, 1, 2, 2) + 1
        Else
          CNT(3, 1, 3, 0) = CNT(3, 1, 3, 0) + 1
          If S(1) < LSL(1) Then CNT(3, 1, 3, 1) = CNT(3, 1, 3, 1) + 1
          If S(1) > USL(1) Then CNT(3, 1, 3, 2) = CNT(3, 1, 3, 2) + 1
        End If
      ElseIf S(3) > USL(3) Then
        CNT(3, 2, 0, 0) = CNT(3, 2, 0, 0) + 1
        If S(2) < LSL(2) Then
          CNT(3, 2, 1, 0) = CNT(3, 2, 1, 0) + 1
          If S(1) < LSL(1) Then CNT(3, 2, 1, 1) = CNT(3, 2, 1, 1) + 1
          If S(1) > USL(1) Then CNT(3, 2, 1, 2) = CNT(3, 2, 1, 2) + 1
        ElseIf S(2) > USL(2) Then
          CNT(3, 2, 2, 0) = CNT(3, 2, 2, 0) + 1
          If S(1) < LSL(1) Then CNT(3, 2, 2, 1) = CNT(3, 2, 2, 1) + 1
          If S(1) > USL(1) Then CNT(3, 2, 2, 2) = CNT(3, 2, 2, 2) + 1
        Else
          CNT(3, 2, 3, 0) = CNT(3, 2, 3, 0) + 1
          If S(1) < LSL(1) Then CNT(3, 2, 3, 1) = CNT(3, 2, 3, 1) + 1
          If S(1) > USL(1) Then CNT(3, 2, 3, 2) = CNT(3, 2, 3, 2) + 1
        End If
      Else
        CNT(3, 3, 0, 0) = CNT(3, 3, 0, 0) + 1
        If S(2) < LSL(2) Then
          CNT(3, 3, 1, 0) = CNT(3, 3, 1, 0) + 1
          If S(1) < LSL(1) Then CNT(3, 3, 1, 1) = CNT(3, 3, 1, 1) + 1
          If S(1) > USL(1) Then CNT(3, 3, 1, 2) = CNT(3, 3, 1, 2) + 1
        ElseIf S(2) > USL(2) Then
          CNT(3, 3, 2, 0) = CNT(3, 3, 2, 0) + 1
```

```
            If S(1) < LSL(1) Then CNT(3, 3, 2, 1) = CNT(3, 3, 2, 1) + 1
            If S(1) > USL(1) Then CNT(3, 3, 2, 2) = CNT(3, 3, 2, 2) + 1
         Else
            CNT(3, 3, 3, 0) = CNT(3, 3, 3, 0) + 1
            If S(1) < LSL(1) Then CNT(3, 3, 3, 1) = CNT(3, 3, 3, 1) + 1
            If S(1) > USL(1) Then CNT(3, 3, 3, 2) = CNT(3, 3, 3, 2) + 1
         End If
      End If
   End If

105 Next SIM
   MPS(K1) = CNT(0, 0, 0, 1) + CNT(0, 0, 0, 2)

   If NV = 1 Then GoTo 110

   MPS(K1) = CNT(0, 0, 1, 0) + CNT(0, 0, 2, 0) + _
           CNT(0, 0, 3, 1) + CNT(0, 0, 3, 2)

   If NV = 2 Then GoTo 110

   MPS(K1) = CNT(0, 1, 0, 0) + CNT(0, 2, 0, 0) + _
           CNT(0, 3, 1, 0) + CNT(0, 3, 2, 0) + _
           CNT(0, 3, 3, 1) + CNT(0, 3, 3, 2)

   If NV = 3 Then GoTo 110

   MPS(K1) = CNT(1, 0, 0, 0) + CNT(2, 0, 0, 0) + _
           CNT(3, 1, 0, 0) + CNT(3, 2, 0, 0) + _
           CNT(3, 3, 1, 0) + CNT(3, 3, 2, 0) + _
           CNT(3, 3, 3, 1) + CNT(3, 3, 3, 2)

110 If MPS(K1) > 0 Then
CPTOT(K1) = (Excel.Application.WorksheetFunction.NormSInv(1 - (MPS(K1) / _
              2000000))) / 3
Else: CPTOT(K1) = 2: End If

Next K1 'K1 GOES FROM 1 TO 3 (LCL, MPS, UCL)

125 'SKIPPING THE SIMULATION LINE NUMBER

'************** PROCESS CAPABILITY INDEX CALCULATION **************

CPA(1, 8) = MIN3VAL(CPA(1, 1), CPA(1, 2), CPA(1, 3))
If CPA(1, 8) < 0 Then CPA(1, 8) = 0
CPA(1, 4) = CPA(1, 4)
If CPA(1, 4) > 2 Then CPA(1, 4) = 2
```

```
CPA(1, 9) = MAX3VAL(CPA(1, 5), CPA(1, 6), CPA(1, 7))
If CPA(1, 9) < CPA(1, 4) Then CPA(1, 9) = CPA(1, 4) + Abs(CPA(1, 4) - CPA(1, 8))
If CPA(1, 9) > 2 Then CPA(1, 9) = 2
MCPA(1) = CPA(1, 8)
MCPA(2) = CPA(1, 4)
MCPA(3) = CPA(1, 9)

If NV = 1 Then GoTo 130

CPA(2, 8) = MIN3VAL(CPA(2, 1), CPA(2, 2), CPA(2, 3))
If CPA(2, 8) < 0 Then CPA(2, 8) = 0
CPA(2, 4) = CPA(2, 4)
If CPA(2, 4) > 2 Then CPA(2, 4) = 2
CPA(2, 9) = MAX3VAL(CPA(2, 5), CPA(2, 6), CPA(2, 7))
If CPA(2, 9) < CPA(2, 4) Then CPA(2, 9) = CPA(2, 4) + Abs(CPA(2, 4) - CPA(2, 8))
If CPA(2, 9) > 2 Then CPA(2, 9) = 2
MCPA(1) = CPTOT(1)
If MCPA(1) < 0 Then MCPA(1) = 0
MCPA(2) = CPTOT(2)
If MCPA(2) > 2 Then MCPA(2) = 2
MCPA(3) = CPTOT(3)
If MCPA(3) < MCPA(2) Then MCPA(3) = MCPA(2) + Abs(MCPA(2) - MCPA(1))
If MCPA(3) > 2 Then MCPA(3) = 2

If NV = 2 Then GoTo 130

CPA(3, 8) = MIN3VAL(CPA(3, 1), CPA(3, 2), CPA(3, 3))
If CPA(3, 8) < 0 Then CPA(3, 8) = 0
CPA(3, 4) = CPA(3, 4)
If CPA(3, 4) > 2 Then CPA(3, 4) = 2
CPA(3, 9) = MAX3VAL(CPA(3, 5), CPA(3, 6), CPA(3, 7))
If CPA(3, 9) < CPA(3, 4) Then CPA(3, 9) = CPA(3, 4) + Abs(CPA(3, 4) - CPA(3, 8))
If CPA(3, 9) > 2 Then CPA(3, 9) = 2

If NV = 3 Then GoTo 130

CPA(4, 8) = MIN3VAL(CPA(4, 1), CPA(4, 2), CPA(4, 3))
If CPA(4, 8) < 0 Then CPA(4, 8) = 0
CPA(4, 4) = CPA(4, 4)
If CPA(4, 4) > 2 Then CPA(4, 4) = 2
CPA(4, 9) = MAX3VAL(CPA(4, 5), CPA(4, 6), CPA(4, 7))
If CPA(4, 9) < CPA(4, 4) Then CPA(4, 9) = CPA(4, 4) + Abs(CPA(4, 4) - CPA(4, 8))
If CPA(4, 9) > 2 Then CPA(4, 9) = 2

130 Openfile1 = FreeFile
Open Ofilepath For Output As #Openfile1
```

```
'*********************** FORMATTING OUTPUT ***********************

If NV = 1 Then
Print #Openfile1, Spc(4); "UNIVARIATE STATISTICS"; Tab(34); "VARIABLE # 1"
Print #Openfile1, "-----------------------------"; Tab(34); "------------"
Print #Openfile1, "Selected Johnson Distribution"; Tab(39); DIST(1)
Print #Openfile1, "NS (Number of Samples)"; Tab(35); NS
Print #Openfile1, "LSL (Lower Spec. Limit)"; Tab(35); SIGN(LSL(1)); _
                Format(Abs(LSL(1)), F6)
Print #Openfile1, "USL (Upper Spec. Limit)"; Tab(35); SIGN(USL(1)); _
                Format(Abs(USL(1)), F6)
Print #Openfile1, "E (Xi)"; Tab(35); SIGN(E(1)); Format(Abs(E(1)), F6)
Print #Openfile1, "L (Lambda)"; Tab(35); SIGN(L(1)); Format(Abs(L(1)), F6)
Print #Openfile1, "D (Delta)"; Tab(35); SIGN(D(1)); Format(Abs(D(1)), F6)
Print #Openfile1, "G (Gamma)"; Tab(35); SIGN(G(1)); Format(Abs(G(1)), F6)
Print #Openfile1, "F-Value of K-S Test"; Tab(36); Format(FV(1), "0.00%")
Print #Openfile1, "PL (PPM < LSL)"; Tab(36); Format(PL(1, 4) * 1000000, "#,0")
Print #Openfile1, "PU (PPM > USL)"; Tab(36); Format(PU(1, 4) * 1000000, "#,0")
Print #Openfile1, "P* (Total PPM Out-of-Spec.)"; _
                Tab(36); Format(PS(1, 4) * 1000000, "#,0")
Print #Openfile1, "P* - Lower Confidence Limit"; _
                Tab(36); Format(PS(1, MAXPOS(1)) * 1000000, "#,0")
Print #Openfile1, "P* - Upper Confidence Limit"; _
                Tab(36); Format(PS(1, MINPOS(1)) * 1000000, "#,0")
Print #Openfile1, "-----------------------------"; Tab(34); "------------"
Print #Openfile1, "Cpa (Capability Index)"; Tab(36); Format(MCPA(2), F6)
Print #Openfile1, "Cpa - Lower Confidence Limit"; Tab(36); Format(MCPA(1), F6)
Print #Openfile1, "Cpa - Upper Confidence Limit"; Tab(36); Format(MCPA(3), F6)
Print #Openfile1, "-----------------------------"; Tab(34); "------------"

ElseIf NV = 2 Then
Print #Openfile1, Spc(4); "UNIVARIATE STATISTICS"; Tab(34); "VARIABLE # 1"; _
                Tab(48); "VARIABLE # 2"
Print #Openfile1, "-----------------------------"; Tab(34); "------------"; _
                Tab(48); "------------"
Print #Openfile1, "Selected Johnson Distribution"; Tab(39); DIST(1); _
                Tab(53); DIST(2)
Print #Openfile1, "NS (Number of Samples)"; Tab(35); NS; _
                Tab(49); NS
Print #Openfile1, "LSL (Lower Spec. Limit)"; Tab(35); SIGN(LSL(1)); _
                Format(Abs(LSL(1)), F6); Tab(49); SIGN(LSL(2)); _
                Format(Abs(LSL(2)), F6)
Print #Openfile1, "USL (Upper Spec. Limit)"; Tab(35); SIGN(USL(1)); _
                Format(Abs(USL(1)), F6); Tab(49); SIGN(USL(2)); _
                Format(Abs(USL(2)), F6)
```

```
Print #Openfile1, "E (Xi)"; Tab(35); SIGN(E(1)); Format(Abs(E(1)), F6); _
                  Tab(49); SIGN(E(2)); Format(Abs(E(2)), F6)
Print #Openfile1, "L (Lambda)"; Tab(35); SIGN(L(1)); Format(Abs(L(1)), F6); _
                  Tab(49); SIGN(L(2)); Format(Abs(L(2)), F6)
Print #Openfile1, "D (Delta)"; Tab(35); SIGN(D(1)); Format(Abs(D(1)), F6); _
                  Tab(49); SIGN(D(2)); Format(Abs(D(2)), F6)
Print #Openfile1, "G (Gamma)"; Tab(35); SIGN(G(1)); Format(Abs(G(1)), F6); _
                  Tab(49); SIGN(G(2)); Format(Abs(G(2)), F6)
Print #Openfile1, "F-Value of K-S Test"; Tab(36); Format(FV(1), "0.00%"); _
                  Tab(50); Format(FV(2), "0.00%")
Print #Openfile1, "PL (PPM < LSL)"; Tab(36); Format(PL(1, 4) * 1000000, "#,0"); _
                  Tab(50); Format(PL(2, 4) * 1000000, "#,0")
Print #Openfile1, "PU (PPM > USL)"; Tab(36); Format(PU(1, 4) * 1000000, "#,0"); _
                  Tab(50); Format(PU(2, 4) * 1000000, "#,0")
Print #Openfile1, "P* (Total PPM Out-of-Spec.)"; _
                  Tab(36); Format(PS(1, 4) * 1000000, "#,0"); _
                  Tab(50); Format(PS(2, 4) * 1000000, "#,0")
Print #Openfile1, "P* - Lower Confidence Limit"; _
                  Tab(36); Format(PS(1, MAXPOS(1)) * 1000000, "#,0"); _
                  Tab(50); Format(PS(2, MAXPOS(1)) * 1000000, "#,0")
Print #Openfile1, "P* - Upper Confidence Limit"; _
                  Tab(36); Format(PS(1, MINPOS(1)) * 1000000, "#,0"); _
                  Tab(50); Format(PS(2, MINPOS(1)) * 1000000, "#,0")
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                  Tab(48); "------------"
Print #Openfile1, "Cpa (Capability Index)"; Tab(36); Format(CPA(1, 4), F6); _
                  Tab(50); Format(CPA(2, 4), F6)
Print #Openfile1, "Cpa - Lower Confidence Limit"; Tab(36); Format(CPA(1, 8), F6); _
                  Tab(50); Format(CPA(2, 8), F6)
Print #Openfile1, "Cpa - Upper Confidence Limit"; Tab(36); Format(CPA(1, 9), F6); _
                  Tab(50); Format(CPA(2, 9), F6)
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                  Tab(48); "------------"
Print #Openfile1,
Print #Openfile1, Spc(3); "MULTIVARIATE STATISTICS"; Tab(37); "SYSTEM"; _
                  Tab(49); "CORRELATION"
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                  Tab(49); "COEFFICIENT"
Print #Openfile1, "Multivariate Johnson System"; Tab(39); DIST(1); DIST(2); _
                  Tab(53); "P12"
Print #Openfile1, "MP* (Total PPM Out-of-Spec.)"; _
                  Tab(36); Format(MPS(2), "#,0"); _
                  Tab(49); "-----------"
Print #Openfile1, "MP* - Lower Confidence Limit"; _
                  Tab(36); Format(MPS(3), "#,0"); _
                  Tab(50); SIGN(P12); Format(Abs(P12), F6)
```

```
Print #Openfile1, "MP* - Upper Confidence Limit"; _
                Tab(36); Format(MPS(1), "#,0")
Print #Openfile1, "------------------------------"; Tab(34); "------------"
Print #Openfile1, "MCpa (Capability Index)"; Tab(36); Format(MCPA(2), F6)
Print #Openfile1, "MCpa - Lower Confidence Limit"; Tab(36); Format(MCPA(1), F6)
Print #Openfile1, "MCpa - Upper Confidence Limit"; Tab(36); Format(MCPA(3), F6)
Print #Openfile1, "------------------------------"; Tab(34); "------------"

ElseIf NV = 3 Then
Print #Openfile1, Spc(4); "UNIVARIATE STATISTICS"; Tab(34); "VARIABLE # 1"; _
                Tab(48); "VARIABLE # 2"; _
                Tab(62); "VARIABLE # 3"
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                Tab(48); "------------"; _
                Tab(62); "------------"
Print #Openfile1, "Selected Johnson Distribution"; Tab(39); DIST(1); _
                Tab(53); DIST(2); _
                Tab(67); DIST(3)
Print #Openfile1, "NS (Number of Samples)"; Tab(35); NS; _
                Tab(49); NS; _
                Tab(63); NS
Print #Openfile1, "LSL (Lower Spec. Limit)"; Tab(35); SIGN(LSL(1)); _
                Format(Abs(LSL(1)), F6); Tab(49); SIGN(LSL(2)); _
                Format(Abs(LSL(2)), F6); Tab(63); SIGN(LSL(3)); _
                Format(Abs(LSL(3)), F6)
Print #Openfile1, "USL (Upper Spec. Limit)"; Tab(35); SIGN(USL(1)); _
                Format(Abs(USL(1)), F6); Tab(49); SIGN(USL(2)); _
                Format(Abs(USL(2)), F6); Tab(63); SIGN(USL(3)); _
                Format(Abs(USL(3)), F6)
Print #Openfile1, "E (Xi)"; Tab(35); SIGN(E(1)); Format(Abs(E(1)), F6); _
                Tab(49); SIGN(E(2)); Format(Abs(E(2)), F6); _
                Tab(63); SIGN(E(3)); Format(Abs(E(3)), F6)
Print #Openfile1, "L (Lambda)"; Tab(35); SIGN(L(1)); Format(Abs(L(1)), F6); _
                Tab(49); SIGN(L(2)); Format(Abs(L(2)), F6); _
                Tab(63); SIGN(L(3)); Format(Abs(L(3)), F6)
Print #Openfile1, "D (Delta)"; Tab(35); SIGN(D(1)); Format(Abs(D(1)), F6); _
                Tab(49); SIGN(D(2)); Format(Abs(D(2)), F6); _
                Tab(63); SIGN(D(3)); Format(Abs(D(3)), F6)
Print #Openfile1, "G (Gamma)"; Tab(35); SIGN(G(1)); Format(Abs(G(1)), F6); _
                Tab(49); SIGN(G(2)); Format(Abs(G(2)), F6); _
                Tab(63); SIGN(G(3)); Format(Abs(G(3)), F6)
Print #Openfile1, "F-Value of K-S Test"; Tab(36); Format(FV(1), "0.00%"); _
                Tab(50); Format(FV(2), "0.00%"); _
                Tab(64); Format(FV(3), "0.00%")
Print #Openfile1, "PL (PPM < LSL)"; Tab(36); Format(PL(1, 4) * 1000000, "#,0"); _
                Tab(50); Format(PL(2, 4) * 1000000, "#,0"); _
```

```
                    Tab(64); Format(PL(3, 4) * 1000000, "#,0")
Print #Openfile1, "PU (PPM > USL)"; Tab(36); Format(PU(1, 4) * 1000000, "#,0"); _
                    Tab(50); Format(PU(2, 4) * 1000000, "#,0"); _
                    Tab(64); Format(PU(3, 4) * 1000000, "#,0")
Print #Openfile1, "P* (Total PPM Out-of-Spec.)"; _
                    Tab(36); Format(PS(1, 4) * 1000000, "#,0"); _
                    Tab(50); Format(PS(2, 4) * 1000000, "#,0"); _
                    Tab(64); Format(PS(3, 4) * 1000000, "#,0")
Print #Openfile1, "P* - Lower Confidence Limit"; _
                    Tab(36); Format(PS(1, MAXPOS(1)) * 1000000, "#,0"); _
                    Tab(50); Format(PS(2, MAXPOS(1)) * 1000000, "#,0"); _
                    Tab(64); Format(PS(3, MAXPOS(1)) * 1000000, "#,0")
Print #Openfile1, "P* - Upper Confidence Limit"; _
                    Tab(36); Format(PS(1, MINPOS(1)) * 1000000, "#,0"); _
                    Tab(50); Format(PS(2, MINPOS(1)) * 1000000, "#,0"); _
                    Tab(64); Format(PS(3, MINPOS(1)) * 1000000, "#,0")
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                    Tab(48); "------------"; _
                    Tab(62); "------------"
Print #Openfile1, "Cpa (Capability Index)"; Tab(36); Format(CPA(1, 4), F6); _
                    Tab(50); Format(CPA(2, 4), F6); _
                    Tab(64); Format(CPA(3, 4), F6)
Print #Openfile1, "Cpa - Lower Confidence Limit"; Tab(36); Format(CPA(1, 8), F6); _
                    Tab(50); Format(CPA(2, 8), F6); _
                    Tab(64); Format(CPA(3, 8), F6)
Print #Openfile1, "Cpa - Upper Confidence Limit"; Tab(36); Format(CPA(1, 9), F6); _
                    Tab(50); Format(CPA(2, 9), F6); _
                    Tab(64); Format(CPA(3, 9), F6)
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                    Tab(48); "------------"; _
                    Tab(62); "------------"
Print #Openfile1,
Print #Openfile1, Spc(3); "MULTIVARIATE STATISTICS"; Tab(37); "SYSTEM"; _
                    Tab(49); "CORRELATION"; Tab(62); "COEFFICIENTS"
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                    Tab(49); "-----------"; Tab(62); "------------"
Print #Openfile1, "Multivariate Johnson System"; Tab(38); DIST(1); DIST(2); _
                    DIST(3); _
                    Tab(53); "P12"; Tab(63); SIGN(P12); Format(Abs(P12), F6)
Print #Openfile1, "MP* (Total PPM Out-of-Spec.)"; _
                    Tab(36); Format(MPS(2), "#,0"); _
                    Tab(53); "P13"; Tab(63); SIGN(P13); Format(Abs(P13), F6)
Print #Openfile1, "MP* - Lower Confidence Limit"; _
                    Tab(36); Format(MPS(3), "#,0"); _
                    Tab(53); "P23"; Tab(63); SIGN(P23); Format(Abs(P23), F6)
Print #Openfile1, "MP* - Upper Confidence Limit"; _
```

```
                Tab(36); Format(MPS(1), "#,0")
Print #Openfile1, "------------------------------"; Tab(34); "------------"
Print #Openfile1, "MCpa (Capability Index)"; Tab(36); Format(MCPA(2), F6)
Print #Openfile1, "MCpa - Lower Confidence Limit"; Tab(36); Format(MCPA(1), F6)
Print #Openfile1, "MCpa - Upper Confidence Limit"; Tab(36); Format(MCPA(3), F6)
Print #Openfile1, "------------------------------"; Tab(34); "------------"

Else
Print #Openfile1, Spc(4); "UNIVARIATE STATISTICS"; Tab(34); "VARIABLE # 1"; _
                Tab(48); "VARIABLE # 2"; _
                Tab(62); "VARIABLE # 3"; _
                Tab(76); "VARIABLE # 4"
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                Tab(48); "------------"; _
                Tab(62); "------------"; _
                Tab(76); "------------"
Print #Openfile1, "Selected Johnson Distribution"; Tab(39); DIST(1); _
                Tab(53); DIST(2); _
                Tab(67); DIST(3); _
                Tab(81); DIST(4)
Print #Openfile1, "NS (Number of Samples)"; Tab(35); NS; _
                Tab(49); NS; _
                Tab(63); NS; _
                Tab(77); NS
Print #Openfile1, "LSL (Lower Spec. Limit)"; Tab(35); SIGN(LSL(1)); _
                Format(Abs(LSL(1)), F6); Tab(49); SIGN(LSL(2)); _
                Format(Abs(LSL(2)), F6); Tab(63); SIGN(LSL(3)); _
                Format(Abs(LSL(3)), F6); Tab(77); SIGN(LSL(4)); _
                Format(Abs(LSL(4)), F6)
Print #Openfile1, "USL (Upper Spec. Limit)"; Tab(35); SIGN(USL(1)); _
                Format(Abs(USL(1)), F6); Tab(49); SIGN(USL(2)); _
                Format(Abs(USL(2)), F6); Tab(63); SIGN(USL(3)); _
                Format(Abs(USL(3)), F6); Tab(77); SIGN(USL(4)); _
                Format(Abs(USL(4)), F6)
Print #Openfile1, "E (Xi)"; Tab(35); SIGN(E(1)); Format(Abs(E(1)), F6); _
                Tab(49); SIGN(E(2)); Format(Abs(E(2)), F6); _
                Tab(63); SIGN(E(3)); Format(Abs(E(3)), F6); _
                Tab(77); SIGN(E(4)); Format(Abs(E(4)), F6)
Print #Openfile1, "L (Lambda)"; Tab(35); SIGN(L(1)); Format(Abs(L(1)), F6); _
                Tab(49); SIGN(L(2)); Format(Abs(L(2)), F6); _
                Tab(63); SIGN(L(3)); Format(Abs(L(3)), F6); _
                Tab(77); SIGN(L(4)); Format(Abs(L(4)), F6)
Print #Openfile1, "D (Delta)"; Tab(35); SIGN(D(1)); Format(Abs(D(1)), F6); _
                Tab(49); SIGN(D(2)); Format(Abs(D(2)), F6); _
                Tab(63); SIGN(D(3)); Format(Abs(D(3)), F6); _
                Tab(77); SIGN(D(4)); Format(Abs(D(4)), F6)
```

```
Print #Openfile1, "G (Gamma)"; Tab(35); SIGN(G(1)); Format(Abs(G(1)), F6); _
                  Tab(49); SIGN(G(2)); Format(Abs(G(2)), F6); _
                  Tab(63); SIGN(G(3)); Format(Abs(G(3)), F6); _
                  Tab(77); SIGN(G(4)); Format(Abs(G(4)), F6)
Print #Openfile1, "F-Value of K-S Test"; Tab(36); Format(FV(1), "0.00%"); _
                  Tab(50); Format(FV(2), "0.00%"); _
                  Tab(64); Format(FV(3), "0.00%"); _
                  Tab(78); Format(FV(4), "0.00%")
Print #Openfile1, "PL (PPM < LSL)"; Tab(36); Format(PL(1, 4) * 1000000, "#,0"); _
                  Tab(50); Format(PL(2, 4) * 1000000, "#,0"); _
                  Tab(64); Format(PL(3, 4) * 1000000, "#,0"); _
                  Tab(78); Format(PL(4, 4) * 1000000, "#,0")
Print #Openfile1, "PU (PPM > USL)"; Tab(36); Format(PU(1, 4) * 1000000, "#,0"); _
                  Tab(50); Format(PU(2, 4) * 1000000, "#,0"); _
                  Tab(64); Format(PU(3, 4) * 1000000, "#,0"); _
                  Tab(78); Format(PU(4, 4) * 1000000, "#,0")
Print #Openfile1, "P* (Total PPM Out-of-Spec.)"; _
                  Tab(36); Format(PS(1, 4) * 1000000, "#,0"); _
                  Tab(50); Format(PS(2, 4) * 1000000, "#,0"); _
                  Tab(64); Format(PS(3, 4) * 1000000, "#,0"); _
                  Tab(78); Format(PS(4, 4) * 1000000, "#,0")
Print #Openfile1, "P* - Lower Confidence Limit"; _
                  Tab(36); Format(PS(1, MAXPOS(1)) * 1000000, "#,0"); _
                  Tab(50); Format(PS(2, MAXPOS(1)) * 1000000, "#,0"); _
                  Tab(64); Format(PS(3, MAXPOS(1)) * 1000000, "#,0"); _
                  Tab(78); Format(PS(4, MAXPOS(1)) * 1000000, "#,0")
Print #Openfile1, "P* - Upper Confidence Limit"; _
                  Tab(36); Format(PS(1, MINPOS(1)) * 1000000, "#,0"); _
                  Tab(50); Format(PS(2, MINPOS(1)) * 1000000, "#,0"); _
                  Tab(64); Format(PS(3, MINPOS(1)) * 1000000, "#,0"); _
                  Tab(78); Format(PS(4, MINPOS(1)) * 1000000, "#,0")
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                  Tab(48); "------------"; _
                  Tab(62); "------------"; _
                  Tab(76); "------------"
Print #Openfile1, "Cpa (Capability Index)"; Tab(36); Format(CPA(1, 4), F6); _
                  Tab(50); Format(CPA(2, 4), F6); _
                  Tab(64); Format(CPA(3, 4), F6); _
                  Tab(78); Format(CPA(4, 4), F6)
Print #Openfile1, "Cpa - Lower Confidence Limit"; Tab(36); Format(CPA(1, 8), F6); _
                  Tab(50); Format(CPA(2, 8), F6); _
                  Tab(64); Format(CPA(3, 8), F6); _
                  Tab(78); Format(CPA(4, 8), F6)
Print #Openfile1, "Cpa - Upper Confidence Limit"; Tab(36); Format(CPA(1, 9), F6); _
                  Tab(50); Format(CPA(2, 9), F6); _
                  Tab(64); Format(CPA(3, 9), F6); _
```

```
                 Tab(78); Format(CPA(4, 9), F6)
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                 Tab(48); "------------"; _
                 Tab(62); "------------"; _
                 Tab(76); "------------"
Print #Openfile1,
Print #Openfile1, Spc(3); "MULTIVARIATE STATISTICS"; Tab(37); "SYSTEM"; _
                 Tab(49); "CORRELATION"; Tab(62); "COEFFICIENTS"
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                 Tab(49); "-----------"; Tab(62); "------------"
Print #Openfile1, "Multivariate Johnson System"; Tab(38); DIST(1); DIST(2); _
                 DIST(3); DIST(4); _
                 Tab(53); "P12"; Tab(63); SIGN(P12); Format(Abs(P12), F6)
Print #Openfile1, "MP* (Total PPM Out-of-Spec.)"; _
                 Tab(36); Format(MPS(2), "#,0"); _
                 Tab(53); "P13"; Tab(63); SIGN(P13); Format(Abs(P13), F6)
Print #Openfile1, "MP* - Lower Confidence Limit"; _
                 Tab(36); Format(MPS(3), "#,0"); _
                 Tab(53); "P23"; Tab(63); SIGN(P23); Format(Abs(P23), F6)
Print #Openfile1, "MP* - Upper Confidence Limit"; _
                 Tab(36); Format(MPS(1), "#,0"); _
                 Tab(53); "P14"; Tab(63); SIGN(P14); Format(Abs(P14), F6)
Print #Openfile1, "------------------------------"; Tab(34); "------------"; _
                 Tab(53); "P24"; Tab(63); SIGN(P24); Format(Abs(P24), F6)
Print #Openfile1, "MCpa (Capability Index)"; Tab(36); Format(MCPA(2), F6); _
                 Tab(53); "P34"; Tab(63); SIGN(P34); Format(Abs(P34), F6)
Print #Openfile1, "MCpa - Lower Confidence Limit"; Tab(36); Format(MCPA(1), F6)
Print #Openfile1, "MCpa - Upper Confidence Limit"; Tab(36); Format(MCPA(3), F6)
Print #Openfile1, "------------------------------"; Tab(34); "------------"

End If
Close #Openfile1
DoCmd.Hourglass False
DoCmd.Beep
MsgBox "DATA WRITTEN TO " & Ofilepath
End Sub
```

```
Sub DistFitter(X() As Double, NS As Integer, E As Double, L As Double, _
        D As Double, G As Double, FV As Double, DIST As String)
'**************************************************************************
'* This Visual Basic Code Fits Univariate Sample Data to Johnson Systems and Selects
'* the Best Fit as Determined by a Selection Decision Matrix.
'**************************************************************************
Dim E0 As Double, E1 As Double, E2 As Double, E3 As Double
Dim E4 As Double, E5 As Double, NE As Double, LSE As Double
Dim NL As Double, LSL As Double, L2P As Integer
Dim D1 As Double, D2 As Double, D3 As Double
Dim D4 As Double, D5 As Double, ND As Double, LSD As Double
Dim G1 As Double, G2 As Double, G3 As Double
Dim G4 As Double, G5 As Double, NG As Double, LSG As Double
Dim DIST1 As String, DIST2 As String, DIST3 As String
Dim DIST4 As String, DIST5 As String, NDIST As String, LSDIST As String
Dim FV1 As Double, FV2 As Double, FV3 As Double
Dim FV4 As Double, FV5 As Double, NFV As Double, LSFV As Double
Dim PV As Double, DM As Double, DELTA As Double, ZG As Double
Dim PM1 As Double, PM2 As Double, PM3 As Double, PM4 As Double
Dim PB1 As Double, PG1 As Double, PB2 As Double, PG2 As Double, B2C As Double
Dim START(1 To 4) As Double, MIN(1 To 4) As Double, N As Integer
Dim YNEWLO As Double, REQMIN As Double, STEP(1 To 4) As Double
Dim KONVGE As Integer, ICOUNT As Long, RESTART As Integer, ITER As Long
Dim Z() As Double, OC() As Double, TC() As Double
ReDim Z(1 To NS): ReDim OC(1 To NS): ReDim TC(1 To NS)

Call Stat1(X(), NS, PM1, PM2, PM3, PM4, PB1, PG1, PB2, PG2)
B2C = CB2(PB1)
NDIST = "N"  ' NORMAL
    NL = 1
    NE = 0
    ND = 1 / Sqr(PM2)
    NG = -PM1 / Sqr(PM2)
Call STDZ(NDIST, NS, X(), NG, ND, NE, NL, Z())
Call TCFD(NS, Z(), TC())
Call OCFD(NS, Z(), OC())
DM = DMAX(NS, OC(), TC())
PV = PKS2(NS, DM)
    NFV = 1 - PV

    LSL = 1
If PM3 < 0 Then
    LSDIST = "S"  'SPECIAL - MLE ESTIMATION
    Call SJohnson(NS, X(), PG1, PM1, PM2, LSD, LSE, LSG)
    Call STDZ(LSDIST, NS, X(), LSG, LSD, LSE, LSL, Z())
    Call TCFD(NS, Z(), TC())
```

```
    Call OCFD(NS, Z(), OC())
    DM = DMAX(NS, OC(), TC())
    PV = PKS2(NS, DM)
        LSFV = 1 - PV

    DIST1 = "S1"  'SPECIAL - MLL OPTIMIZATION
    N = 2: START(1) = LSD: START(2) = LSG: START(3) = LSE
    Call NELMIN(DIST1, NS, X(), N, START(), MIN(), YNEWLO, REQMIN, _
                STEP, KONVGE, ICOUNT)
    RESTART = Fix(ICOUNT / 10000): ITER = ICOUNT - RESTART * 10000
        D1 = MIN(1)
        G1 = MIN(2)
    Call STDZ(LSDIST, NS, X(), G1, D1, LSE, LSL, Z())
    Call TCFD(NS, Z(), TC())
    Call OCFD(NS, Z(), OC())
    DM = DMAX(NS, OC(), TC())
    PV = PKS2(NS, DM)
        FV1 = 1 - PV

    If FV1 > LSFV Then
        LSD = D1
        LSG = G1
        LSFV = FV1
    End If

    DIST2 = "S2"  'SPECIAL - MDE OPTIMIZATION
    N = 2: START(1) = LSD: START(2) = LSG: START(3) = LSE
    Call NELMIN(DIST2, NS, X(), N, START(), MIN(), YNEWLO, REQMIN, _
                STEP, KONVGE, ICOUNT)
    RESTART = Fix(ICOUNT / 10000): ITER = ICOUNT - RESTART * 10000
        D2 = MIN(1)
        G2 = MIN(2)
    Call STDZ(LSDIST, NS, X(), G2, D2, LSE, LSL, Z())
    Call TCFD(NS, Z(), TC())
    Call OCFD(NS, Z(), OC())
    DM = DMAX(NS, OC(), TC())
    PV = PKS2(NS, DM)
        FV2 = 1 - PV

    If FV2 > LSFV Then
        LSD = D2
        LSG = G2
        LSFV = FV2
    End If

Else
```

```
    LSDIST = "L"  'LOGNORMAL - 3 PARAMETER MLE
    Call LJohnson(NS, X(), PG1, PM1, PM2, LSD, LSE, LSG, L2P)
    Call STDZ(LSDIST, NS, X(), LSG, LSD, LSE, LSL, Z())
    Call TCFD(NS, Z(), TC())
    Call OCFD(NS, Z(), OC())
    DM = DMAX(NS, OC(), TC())
    PV = PKS2(NS, DM)
        LSFV = 1 - PV
        E0 = LSE

    If L2P = 0 Then GoTo 20 ' Skip the 2-parameter trial

    DIST1 = "L"  'LOGNORMAL - 2 PARAMETER MLE
        E1 = 0
    Call STDZ(DIST1, NS, X(), LSG, LSD, E1, LSL, Z())
    Call TCFD(NS, Z(), TC())
    Call OCFD(NS, Z(), OC())
    DM = DMAX(NS, OC(), TC())
    PV = PKS2(NS, DM)
        FV1 = 1 - PV

    If FV1 > LSFV Then
        LSE = E1
        LSFV = FV1
    End If

20  DIST2 = "L1"  'LOGNORMAL - 3 PARAMETER MLL OPTIMIZATION
        E2 = E0
    N = 2: START(1) = LSD: START(2) = LSG: START(3) = E2
    Call NELMIN(DIST1, NS, X(), N, START(), MIN(), YNEWLO, REQMIN, _
                STEP, KONVGE, ICOUNT)
    RESTART = Fix(ICOUNT / 10000): ITER = ICOUNT - RESTART * 10000
        D2 = MIN(1)
        G2 = MIN(2)
    Call STDZ(LSDIST, NS, X(), G2, D2, E2, LSL, Z())
    Call TCFD(NS, Z(), TC())
    Call OCFD(NS, Z(), OC())
    DM = DMAX(NS, OC(), TC())
    PV = PKS2(NS, DM)
        FV2 = 1 - PV

    If FV2 > LSFV Then
        LSE = E2
        LSD = D2
        LSG = G2
        LSFV = FV2
```

```
End If

If L2P = 0 Then GoTo 30 ' Skip the 2-parameter trial

DIST3 = "L1"  'LOGNORMAL - 2 PARAMETER MLL OPTIMIZATION
     E3 = 0
N = 2: START(1) = LSD: START(2) = LSG: START(3) = E3
Call NELMIN(DIST3, NS, X(), N, START(), MIN(), YNEWLO, REQMIN, _
               STEP, KONVGE, ICOUNT)
RESTART = Fix(ICOUNT / 10000): ITER = ICOUNT - RESTART * 10000
     D3 = MIN(1)
     G3 = MIN(2)
Call STDZ(LSDIST, NS, X(), G3, D3, E3, LSL, Z())
Call TCFD(NS, Z(), TC())
Call OCFD(NS, Z(), OC())
DM = DMAX(NS, OC(), TC())
PV = PKS2(NS, DM)
     FV3 = 1 - PV

If FV3 > LSFV Then
     LSE = E3
     LSD = D3
     LSG = G3
     LSFV = FV3
End If

30   DIST4 = "L2"  'LOGNORMAL - 3 PARAMETER MDE OPTIMIZATION
     E4 = E0
N = 2: START(1) = LSD: START(2) = LSG: START(3) = E4
Call NELMIN(DIST4, NS, X(), N, START(), MIN(), YNEWLO, REQMIN, _
               STEP, KONVGE, ICOUNT)
RESTART = Fix(ICOUNT / 10000): ITER = ICOUNT - RESTART * 10000
     D4 = MIN(1)
     G4 = MIN(2)
Call STDZ(LSDIST, NS, X(), G4, D4, E4, LSL, Z())
Call TCFD(NS, Z(), TC())
Call OCFD(NS, Z, OC)
DM = DMAX(NS, OC(), TC())
PV = PKS2(NS, DM)
     FV4 = 1 - PV

If FV4 > LSFV Then
     LSE = E4
     LSD = D4
     LSG = G4
     LSFV = FV4
```

```
    End If

    If L2P = 0 Then GoTo 40 ' Skip the 2-parameter trial

    DIST5 = "L2"  'LOGNORMAL - 2 PARAMETER MDE OPTIMIZATION
        E5 = 0
    N = 2: START(1) = LSD: START(2) = LSG: START(3) = E5
    Call NELMIN(DIST5, NS, X(), N, START(), MIN(), YNEWLO, REQMIN, _
                STEP, KONVGE, ICOUNT)
    RESTART = Fix(ICOUNT / 10000): ITER = ICOUNT - RESTART * 10000
        D5 = MIN(1)
        G5 = MIN(2)
    Call STDZ(LSDIST, NS, X(), G5, D5, E5, LSL, Z())
    Call TCFD(NS, Z(), TC())
    Call OCFD(NS, Z(), OC())
    DM = DMAX(NS, OC(), TC())
    PV = PKS2(NS, DM)
        FV5 = 1 - PV

    If FV5 > LSFV Then
        LSE = E5
        LSD = D5
        LSG = G5
        LSFV = FV5
    End If

40  End If

DELTA = LSFV - NFV
ZG = Abs(PG1) / Sqr(6 / NS)

If NFV > 0.2 And DELTA < 0.3 And ZG < 1.96 Then  'CALL IT NORMAL
    E = NE
    L = NL
    D = ND
    G = NG
    FV = NFV
    DIST = NDIST
    GoTo 50
ElseIf LSFV > 0.2 And ZG > 1.96 Then  'CALL IT SPECIAL OR LOGNORMAL
    E = LSE
    L = LSL
    D = LSD
    G = LSG
    FV = LSFV
    DIST = LSDIST
```

```
    GoTo 50
End If

REQMIN = 1E-16: KONVGE = 5: ICOUNT = 1000
STEP(1) = 1: STEP(2) = 1: STEP(3) = 1: STEP(4) = 1

If B2C < PB2 Then
    DIST = "U"  'UNBOUNDED - MDE OPTIMIZATION
    Call UJohnson(PB1, PB2, PM1, PM2, PM3, E, L, G, D)
    N = 4: START(1) = D: START(2) = E: START(3) = G: START(4) = L
    Call NELMIN(DIST, NS, X(), N, START(), MIN(), YNEWLO, REQMIN, _
                STEP, KONVGE, ICOUNT)
    RESTART = Fix(ICOUNT / 10000): ITER = ICOUNT - RESTART * 10000
        D = MIN(1)
        E = MIN(2)
        G = MIN(3)
        L = MIN(4)
    Call STDZ(DIST, NS, X(), G, D, E, L, Z())
    Call TCFD(NS, Z(), TC())
    Call OCFD(NS, Z(), OC())
    DM = DMAX(NS, OC(), TC())
    PV = PKS2(NS, DM)
        FV = 1 - PV
Else
    DIST = "B"  'BOUNDED - MDE OPTIMIZATION
    Call BJohnson(NS, X(), PM3, D, E, G, L)
    N = 2: START(1) = D: START(2) = G: START(3) = E: START(4) = L
    Call NELMIN(DIST, NS, X(), N, START(), MIN(), YNEWLO, REQMIN, _
                STEP, KONVGE, ICOUNT)
    RESTART = Fix(ICOUNT / 10000): ITER = ICOUNT - RESTART * 10000
        D = MIN(1)
        G = MIN(2)
    Call STDZ(DIST, NS, X(), G, D, E, L, Z())
    Call TCFD(NS, Z(), TC())
    Call OCFD(NS, Z(), OC())
    DM = DMAX(NS, OC(), TC())
    PV = PKS2(NS, DM)
        FV = 1 - PV
End If

50  End Sub
```

```
Sub Stat1(X() As Double, N As Integer, M1 As Double, M2 As Double, _
        M3 As Double, M4 As Double, B1 As Double, G1 As Double, _
        B2 As Double, G2 As Double)
'************************************************************************
'* This Visual Basic Code Calculates the Descriptive Statistics for the Sample Data.
'************************************************************************
Dim Sum1 As Double, Sum2 As Double, Sum3 As Double, Sum4 As Double
Dim I As Integer

Sum1 = 0: Sum2 = 0: Sum3 = 0: Sum4 = 0
For I = 1 To N
  Sum1 = Sum1 + X(I)
Next I
M1 = Sum1 / N '1st Moment about Origin (Sample Average)
For I = 1 To N
  Sum2 = Sum2 + (X(I) - M1) ^ 2
  Sum3 = Sum3 + (X(I) - M1) ^ 3
  Sum4 = Sum4 + (X(I) - M1) ^ 4
Next I
M2 = Sum2 / N '2nd Moment about Mean (Sample Variance)
M3 = Sum3 / N '3rd Moment about Mean
M4 = Sum4 / N '4th Moment about Mean
B1 = M3 ^ 2 / M2 ^ 3 'Skewness Measure
G1 = Sgn(M3) * Sqr(B1) 'Skewness Measure
B2 = M4 / M2 ^ 2 'Kurtosis Measure
G2 = B2 - 3 'Kurtosis Measure
End Sub
```

```
Sub UJohnson(B1 As Double, B2 As Double, M1 As Double, M2 As Double, _
             M3 As Double, E As Double, L As Double, G As Double, D As Double)
'************************************************************************
'* This Visual Basic Code Calculates Initial Parameter Estimates for the Johnson
'* Unbounded System on Sample Data.
'************************************************************************
Dim W As Double, M As Double, A0 As Double, A1 As Double, A2 As Double
Dim A3 As Double, K As Double, XB As Double, I As Long, C As Double
Dim B As Double, B1H As Double, B1D As Double, S As Double

I = 0
B1D = 10
D = 1 'changed from D = 4, due to problems with M calculation
W = Exp(1 / D ^ 2)
Do Until B1D <= 0.00000001
K = 2 * (B2 - 3) / (W - 1)
A0 = W ^ 5 + 3 * W ^ 4 + 6 * W ^ 3 + 10 * W ^ 2 + 9 * W + 3
A1 = 8 * (W ^ 4 + 3 * W ^ 3 + 6 * W ^ 2 + 7 * W + 3)
A2 = 8 * (W ^ 3 + 3 * W ^ 2 + 6 * W + 6)
A3 = 4 * (W + 1) * K - A1
M = (A3 + Sqr(A3 ^ 2 - 4 * (A2 - 4 * K) * (A0 - (W + 1) ^ 2 * K))) / _
    (2 * (A2 - 4 * K))
B1H = (M * (W - 1) * (4 * (W + 2) * M + 3 * (W + 1) ^ 2) ^ 2) / _
      (2 * (2 * M + W + 1) ^ 3)
B1D = Abs(B1 - B1H)
B = B1 / B1H
C = 3 - 2 * B2 * (1 - B) - B * (W ^ 4 + 2 * W ^ 2 + 3)
W = Sqr(-1 + Sqr(1 - C))
I = I + 1
Loop
D = Sqr(1 / Log(W))
G = -Sgn(M3) * Abs(D * ArcSinh(Sqr(M)))
S = Sqr(M2)
L = S / Sqr(0.5 * (W - 1) * (W * Cosh(2 * G / D) + 1))
XB = M1
E = XB + L * Sqr(W) * Sinh(G / D)
End Sub
```

```
Sub LJohnson(M As Integer, X() As Double, G1 As Double, M1 As Double, _
             M2 As Double, D As Double, E As Double, G As Double, _
             L2P As Integer)
'****************************************************************************
'* This Visual Basic Code Calculates Initial Parameter Estimates for the Johnson
'* Lognormal System on Sample Data.
'****************************************************************************
Dim T As Double, T1 As Double, T2 As Double, S As Double, W As Double
Dim MINX As Double, MAXX As Double, MEDX As Double
Dim Sum1 As Double, Sum2 As Double, I As Integer
Dim EM As Double, EE As Double, EB As Double

T1 = (((G1 / 2) + Sqr((G1 ^ 2 / 4) + 1)) ^ (1 / 3))
T2 = -((Abs((G1 / 2) - Sqr((G1 ^ 2 / 4) + 1))) ^ (1 / 3))
T = T1 + T2
S = Sqr(M2)
EM = M1 - (S / T)
MINX = Excel.Application.WorksheetFunction.MIN(X)
MAXX = Excel.Application.WorksheetFunction.MAX(X)
MEDX = Excel.Application.WorksheetFunction.Median(X)
' Check for 2-parameter trial necessary
If MINX > 0 Then L2P = 1 Else: L2P = 0
EE = (MINX * MAXX - MEDX ^ 2) / (MINX + MAXX - 2 * MEDX)
EB = MINX - (MAXX - MINX) / M
' JOHNSON MLE WITH LOCATION KNOWN
If EM >= MINX Or EM < (EB - 3 * (MAXX - MINX)) Then EM = EB
If EE >= MINX Or EE < EB - (MAXX - MINX) Then EE = EB
If EM < EE Then
   E = EM
Else
   E = EE
End If
Sum1 = 0: Sum2 = 0
For I = 1 To M
   Sum1 = Sum1 + Log(X(I) - E)
Next I
T1 = Sum1 / M
For I = 1 To M
   Sum2 = Sum2 + (Log(X(I) - E) - T1) ^ 2
Next I
T2 = Sqr(Sum2 / M)
D = 1 / T2
G = -D * T1
End Sub
```

```
Sub SJohnson(M As Integer, X() As Double, G1 As Double, M1 As Double, _
             M2 As Double, D As Double, E As Double, G As Double)
'*************************************************************************
'* This Visual Basic Code Calculates Initial Parameter Estimates for the Johnson
'* - Lewis Special System on Sample Data.
'*************************************************************************
Dim T As Double, T1 As Double, T2 As Double, S As Double, W As Double
Dim MINX As Double, MAXX As Double, MEDX As Double
Dim Sum1 As Double, Sum2 As Double, I As Integer
Dim EM As Double, EE As Double, EB As Double

T1 = (((G1 / 2) + Sqr((G1 ^ 2 / 4) + 1)) ^ (1 / 3))
T2 = -((Abs((G1 / 2) - Sqr((G1 ^ 2 / 4) + 1))) ^ (1 / 3))
T = T1 + T2
S = Sqr(M2)
EM = M1 - (S / T)
MINX = Excel.Application.WorksheetFunction.MIN(X)
MAXX = Excel.Application.WorksheetFunction.MAX(X)
MEDX = Excel.Application.WorksheetFunction.Median(X)
EE = (Abs(MINX) * Abs(MAXX) + Abs(MEDX) ^ 2) / (MINX + MAXX + 2 * MEDX)
EB = MAXX + (MAXX - MINX) / M
' JOHNSON MLE WITH LOCATION KNOWN
If EM <= MAXX Or EM > (EB + 3 * (MAXX - MINX)) Then EM = EB
If EE <= MAXX Or EM > (EB + 3 * (MAXX - MINX)) Then EE = EB
If EM > EE Then
   E = EM
Else
   E = EE
End If

Sum1 = 0: Sum2 = 0
For I = 1 To M
   Sum1 = Sum1 + Log(E - X(I))
Next I
T1 = Sum1 / M
For I = 1 To M
   Sum2 = Sum2 + (Log(E - X(I)) - T1) ^ 2
Next I
T2 = Sqr(Sum2 / M)
D = 1 / T2
G = -D * T1
End Sub
```

```
Sub BJohnson(M As Integer, X() As Double, M3 As Double, D As Double, _
            E As Double, G As Double, L As Double)
'*********************************************************************
'* This Visual Basic Code Calculates Initial Parameter Estimates for the Johnson
'* Bounded System on Sample Data.
'*********************************************************************
Dim MINX As Double, MAXX As Double, X1 As Double, X3 As Double
Dim Q1 As Double, Q3 As Double, K1 As Double, K3 As Double

MINX = Excel.Application.WorksheetFunction.MIN(X)
MAXX = Excel.Application.WorksheetFunction.MAX(X)
Q1 = Excel.Application.WorksheetFunction.Quartile(X, 1)
Q3 = Excel.Application.WorksheetFunction.Quartile(X, 3)
X1 = Excel.Application.WorksheetFunction.NormSInv(0.25)
X3 = Excel.Application.WorksheetFunction.NormSInv(0.75)
E = MINX - (1 / M) * (MAXX - MINX)
L = MAXX + (1 / M) * (MAXX - MINX) - E
K1 = Log((Q1 - E) / (E + L - Q1))
K3 = Log((Q3 - E) / (E + L - Q3))
D = (X1 - X3) / (K1 - K3)
G = Sgn(M3) * Abs(X1 - D * K1)
End Sub
```

```
Function PKS2(N As Integer, D As Double) As Double
'***********************************************************************
'* This Visual Basic Code Represents Algorithm 487 (CACM) by John Pomeranz.
'* The algorithm calculates the exact cumulative distribution of the two-sided
'* Kolmogorov-Smirnov statistics for samples with few observations.
'***********************************************************************
Dim Q(1 To 141) As Double, FACT(1 To 141) As Double, SUM As Double
Dim FT As Double, FU As Double, FV As Double, FN As Double, FND As Double
Dim NDT As Integer, ND As Integer, NDD As Integer, NDP As Integer
Dim CI As Double, NDDP As Integer, SIGN As Integer
Dim I As Integer, JMAX As Integer, K As Integer, J As Integer
'***********************************************************************
'* N IS THE SAMPLE SIZE USED.
'* D IS THE MAXIMUM MAGNITUDE (OF THE DISCREPANCY
'* BETWEEN THE EMPIRICAL AND PROPOSED DISTRIBUTIONS)
'* IN EITHER THE POSITIVE OR NEGATIVE DIRECTION.
'* PKS2 IS THE EXACT PROBABILITY OF OBTAINING A
'* DEVIATION NO LARGER THAN D.
'* THESE FORMULAS APPEAR AS (23) AND (24) IN
'* J. DURBIN. THE PROBABILITY THAT THE SAMPLE
'* DISTRIBUTION FUNCTION LIES BETWEEN TWO PARALLEL
'* STRAIGHT LINES. ANNALS OF MATHEMATICAL STATISTICS
'* 39, 2(APRIL 1968), 398-411.
'***********************************************************************

If N = 1 Then GoTo 90
FN = CDbl(N)
FND = FN * D
NDT = CInt(Fix(2 * FND))
If NDT < 1 Then GoTo 100
ND = CInt(Fix(FND))
NDD = Excel.Application.WorksheetFunction.MIN(2 * ND, N)
NDP = ND + 1
NDDP = NDD + 1
FACT(1) = 1
CI = 1
For I = 1 To N
   FACT(I + 1) = FACT(I) * CI
   CI = CI + 1
Next I
Q(1) = 1
If NDD = 0 Then GoTo 50
CI = 1
For I = 1 To NDD
   Q(I + 1) = CI ^ I / FACT(I + 1)
   CI = CI + 1
```

```
Next I
If NDP > N Then GoTo 80
FV = CDbl(NDP) - FND
JMAX = CInt(Int(FV)) + 1
For I = NDP To NDD
   SUM = 0
   FT = FND
   K = I
   FU = FV
   For J = 1 To JMAX
      SUM = SUM + FT ^ (J - 2) / FACT(J) * FU ^ K / FACT(K + 1)
      FT = FT + 1
      FU = FU - 1
      K = K - 1
   Next J
   Q(I + 1) = Q(I + 1) - 2 * FND * SUM
   JMAX = JMAX + 1
   FV = FV + 1
Next I
If NDD = N Then GoTo 80
50  For I = NDDP To N
   SUM = 0
   SIGN = 1
   FT = 2 * FND
   For J = 1 To NDT
      FT = FT - 1
      K = I - J + 1
      SUM = SUM + SIGN * FT ^ J / FACT(J + 1) * Q(K)
      SIGN = -SIGN
   Next J
   Q(I + 1) = SUM
Next I
80  PKS2 = Q(N + 1) * FACT(N + 1) / FN ^ N
   GoTo 110
90  PKS2 = 2 * D - 1
   GoTo 110
100 PKS2 = 0
110 If PKS2 < 0.000001 Then PKS2 = 0
   If PKS2 > 0.999999 Then PKS2 = 1
End Function
```

```
Function DMAX(NV As Integer, OC() As Double, TC() As Double) As Double
'*************************************************************************
'* This Visual Basic Code Calculates the Kolmogorov-Smirnov D-Statistic,
'* Which Represents the Largest Deviation Between the Observed Cumulative
'* Frequency Distribution and the Expected Cumulative Frequency Distribution.
'*************************************************************************
Dim I As Integer, D As Double

DMAX = 0
For I = 1 To NV
   D = Abs(TC(I) - OC(I))
   If D > DMAX Then DMAX = D
Next I
End Function


Sub OCFD(NV As Integer, X() As Double, OC() As Double)
'*************************************************************************
'* This Visual Basic Code Calculates the Observed Cumulative Frequency Distribution.
'*************************************************************************
Dim COUNT As Integer, XP As Double, I As Integer, J As Integer

For I = 1 To NV
   XP = X(I)
   COUNT = 0
   For J = 1 To NV
      If XP >= X(J) Then COUNT = COUNT + 1
   Next J
   OC(I) = CDbl(COUNT) / CDbl(NV)
Next I
End Sub


Sub TCFD(NV As Integer, Z() As Double, TC() As Double)
'*************************************************************************
'* This Visual Basic Code Calculates the Expected Cumulative Frequency Distribution.
'* The Expected of "Theoretical" Distribution is always the Standard Normal.
'*************************************************************************
Dim I As Integer

For I = 1 To NV
   TC(I) = Excel.Application.WorksheetFunction.NormSDist(Z(I))
Next I
End Sub
```

```
Sub STDZ(DIST As String, NV As Integer, H() As Double, G As Double, _
            D As Double, E As Double, L As Double, Z() As Double)
'*************************************************************************
'*  This Visual Basic Code Calculates the Z Value of the Given Johnson System.
'*  The Z-Value is the Standard Normal Transformation.
'*************************************************************************
Dim I As Integer

Select Case DIST
Case "L"  'LOGNORMAL
    For I = 1 To NV
      Z(I) = G + D * Log(H(I) - E)
    Next I
Case "S"  'SPECIAL
    For I = 1 To NV
      Z(I) = G + D * Log(E - H(I))
    Next I
Case "U"  'UNBOUNDED
    For I = 1 To NV
      Z(I) = G + D * ArcSinh((H(I) - E) / L)
    Next I
Case "B"  'BOUNDED
    For I = 1 To NV
      Z(I) = G + D * Log((H(I) - E) / (E + L - H(I)))
    Next I
Case "N"  'NORMAL
    For I = 1 To NV
      Z(I) = G + D * H(I)
    Next I
End Select
End Sub
```

```
Sub NELMIN(DIST As String, M As Integer, X() As Double, N As Integer, _
            START() As Double, MIN() As Double, YNEWLO As Double, _
            REQMIN As Double, STEP() As Double, KONVGE As Integer, _
            ICOUNT As Long)
'**********************************************************************
'* This Visual Basic Code Represents Algorithm AS 47 by R. O'Neill.
'* ALGORITHM AS 47  APPLIED STATISTICS (J.R.STATIST.SOC C),
'* (1971) VOL.20, NO.3
'*
'* THE NELDER-MEAD SIMPLEX MINIMIZATION PROCEDURE
'*
'* PURPOSE :: TO FIND THE MINIMUM VALUE OF A USER-SPECIFIED
'* FUNCTION.
'*
'* REFERENCE :: NELDER,J.A. AND MEAD,R. (1965).  A SIMPLEX METHOD
'* FOR FUNCTION MINIMIZATION.  COMPUTER J.,VOL.7,308-313
'**********************************************************************
Dim KCOUNT As Long, JCOUNT As Long, DN As Double, NN As Integer
Dim DEL As Double, I As Integer, P(20, 21) As Double, Z As Double
Dim SUM As Double, SUMM As Double, J As Integer, YLO As Double
Dim IHI As Integer, PBAR(20) As Double, PSTAR(20) As Double
Dim RCOEFF As Double, ECOEFF As Double, CCOEFF As Double
Dim Y2STAR As Double, L As Integer, CURMIN As Double, DNN As Double
Dim Y(20) As Double, ILO As Integer, P2STAR(20) As Double, YSTAR As Double
'**********************************************************************
'* FORMAL PARAMETERS ::
'*
'* N        : INPUT  :  THE NUMBER OF VARIABLES OVER WHICH WE
'*                   :  ARE MINIMIZING.
'* START    : INPUT  :  ARRAY;  CONTAINS THE COORDINATES OF THE
'*                   :  STARTING POINT.
'* MIN      : OUTPUT :  ARRAY;  CONTAINS THE COORDINATES OF THE
'*                   :  MINIMUM.
'* YNEWLO   : OUTPUT :  THE MINIMUM VALUE OF THE FUNCTION.
'* REQMIN   : INPUT  :  THE TERMINATING LIMIT FOR THE VARIANCE
'*                   :  OF FUNCTION VALUES.
'* STEP     : INPUT  :  ARRAY; DETERMINES THE SIZE AND SHAPE OF
'*                   :  THE INITIAL SIMPLEX.  THE RELATIVE
'*                   :  MAGNITUDES OF ITS N ELEMENTS SHOULD
'*                   :  REFLECT THE UNITS OF THE N VARIABLES.
'* KONVGE   : INPUT  :  THE CONVERGENCE CHECK IS CARRIED OUT
'*                   :  EVERY KONVGE ITERATIONS.
'* ICOUNT   : INPUT  :  MAXIMUM NUMBER OF FUNCTION
'*                   :  EVALUATIONS.
'*            OUTPUT :  FUNCTION EVALUATIONS PERFORMED + 10,000
'*                   :  TIMES NUMBER OF RESTARTS.  NEGATIVE
```

```
'*                          :    ICOUNT VALUE IDENTIFIES INPUT PARAMETER
'*                          :    FAULT(S).
'*
'* ALL VARIABLES AND ARRAYS ARE TO BE DECLARED IN THE CALLING
'* PROGRAM AS DOUBLE PRECISION.
'*
'* AUXILIARY ALGORITHM :: THE DOUBLE PRECISION FUNCTION
'* SUBPROGRAM FN(A) CALCULATES THE FUNCTION VALUE AT POINT A.
'* A IS DOUBLE PRECISION WITH N ELEMENTS.
'***********************************************************************

RCOEFF = 1
ECOEFF = 2
CCOEFF = 0.5

KCOUNT = ICOUNT
ICOUNT = 0
If REQMIN <= 0 Then ICOUNT = ICOUNT - 1
If N > 20 Then ICOUNT = ICOUNT - 10
If KONVGE <= 0 Then ICOUNT = ICOUNT - 100
If ICOUNT < 0 Then GoTo 2000

JCOUNT = KONVGE
DN = CDbl(N)
NN = N + 1
DNN = CDbl(NN)
DEL = 1

1001    For I = 1 To N
P(I, NN) = START(I)
Next I
Z = FN(DIST, M, X(), START)
Y(NN) = Z
SUM = Z
SUMM = Z * Z
For J = 1 To N
START(J) = START(J) + STEP(J) * DEL
For I = 1 To N
P(I, J) = START(I)
Next I
Z = FN(DIST, M, X(), START)
Y(J) = Z
SUM = SUM + Z
SUMM = SUMM + Z * Z
START(J) = START(J) - STEP(J) * DEL
Next J
```

```
1000    YLO = Y(1)
YNEWLO = YLO
ILO = 1
IHI = 1
For I = 2 To NN
If Y(I) >= YLO Then GoTo 4
YLO = Y(I)
ILO = I
4       If Y(I) <= YNEWLO Then GoTo 5
YNEWLO = Y(I)
IHI = I
5       Next I
SUM = SUM - YNEWLO
SUMM = SUMM - YNEWLO * YNEWLO

For I = 1 To N
Z = 0
For J = 1 To NN
Z = Z + P(I, J)
Next J
Z = Z - P(I, IHI)
PBAR(I) = Z / DN
Next I

For I = 1 To N
PSTAR(I) = (1 + RCOEFF) * PBAR(I) - RCOEFF * P(I, IHI)
Next I
YSTAR = FN(DIST, M, X(), PSTAR)
ICOUNT = ICOUNT + 1
If YSTAR >= YLO Then GoTo 12

For I = 1 To N
P2STAR(I) = ECOEFF * PSTAR(I) + (1 - ECOEFF) * PBAR(I)
Next I
Y2STAR = FN(DIST, M, X(), P2STAR)
ICOUNT = ICOUNT + 1

If Y2STAR >= YLO Then GoTo 19
10      For I = 1 To N
P(I, IHI) = P2STAR(I)
Next I
Y(IHI) = Y2STAR
GoTo 900

12      L = 0
```

```
For I = 1 To NN
If Y(I) > YSTAR Then L = L + 1
Next I
If L > 1 Then GoTo 19
If L = 0 Then GoTo 15

For I = 1 To N
P(I, IHI) = PSTAR(I)
Next I
Y(IHI) = YSTAR

15      For I = 1 To N
P2STAR(I) = CCOEFF * P(I, IHI) + (1 - CCOEFF) * PBAR(I)
Next I
Y2STAR = FN(DIST, M, X(), P2STAR)
ICOUNT = ICOUNT + 1
If Y2STAR <= Y(IHI) Then GoTo 10

SUM = 0
SUMM = 0
For J = 1 To NN
For I = 1 To N
P(I, J) = (P(I, J) + P(I, ILO)) * 0.5
MIN(I) = P(I, J)
Next I
Y(J) = FN(DIST, M, X(), MIN)
SUM = SUM + Y(J)
SUMM = SUMM + Y(J) * Y(J)
Next J
ICOUNT = ICOUNT + NN
GoTo 901

19  For I = 1 To N
P(I, IHI) = PSTAR(I)
Next I
Y(IHI) = YSTAR
900     SUM = SUM + Y(IHI)
SUMM = SUMM + Y(IHI) * Y(IHI)
901     JCOUNT = JCOUNT - 1
If JCOUNT <> 0 Then GoTo 1000

If ICOUNT > KCOUNT Then GoTo 22
JCOUNT = KONVGE
CURMIN = (SUMM - (SUM * SUM) / DNN) / DN

If CURMIN >= REQMIN Then GoTo 1000
```

```
22      For I = 1 To N
MIN(I) = P(I, IHI)
Next I
YNEWLO = Y(IHI)
If ICOUNT > KCOUNT Then GoTo 2000
For I = 1 To N
DEL = STEP(I) * 0.001
MIN(I) = MIN(I) + DEL
Z = FN(DIST, M, X(), MIN)
If Z < YNEWLO Then GoTo 25
MIN(I) = MIN(I) - DEL - DEL
Z = FN(DIST, M, X(), MIN)
If Z < YNEWLO Then GoTo 25
MIN(I) = MIN(I) + DEL
Next I
GoTo 2000

25      For I = 1 To N
START(I) = MIN(I)
Next I
DEL = 0.001
ICOUNT = ICOUNT + 10000
GoTo 1001
2000    End Sub
```

```
Function FN(DIST As String, M As Integer, X() As Double, A() As Double) As Double
'* THIS IS THE FUNCTION FOR NELMIN

Dim I As Integer, Y As Double, Y1 As Double, Y2 As Double, YTOT As Double
Dim XMIN As Double, XMAX As Double, E As Double, L As Double
Dim Z() As Double, DM As Double, TC() As Double

Static OC() As Double: Static E1 As Double: Static L1 As Double
ReDim OC(1 To M): ReDim TC(1 To M): ReDim Z(1 To M)
Const PI As Double = 3.14159265358979

Select Case DIST
Case "L1"  'LOGNORMAL MLL OPTIMIZED
   YTOT = 1
   If A(3) = 0 Then A(3) = E1 Else E1 = A(3)
   For I = 1 To M
      Y1 = A(1) / (Sqr(2 * PI) * X(I))
      Y2 = (A(2) + A(1) * Log(X(I) - A(3))) ^ 2
      Y = Y1 * Exp(-0.5 * Y2)
      YTOT = YTOT * Y
   Next I
If YTOT > 0 Then FN = -Log(YTOT) Else FN = 1000000#

Case "L2"  'LOGNORMAL MDE OPTIMIZED
   If A(3) = 0 Then A(3) = E1 Else E1 = A(3)
   For I = 1 To M
      Z(I) = A(2) + A(1) * Log(X(I) - A(3))
   Next I
   Call TCFD(M, Z(), TC())
   If OC(1) = 0 Then
      Call OCFD(M, Z(), OC())
   End If
FN = DMAX(M, OC(), TC())

Case "S1"  'SPECIAL MLL OPTIMIZED
   YTOT = 1
   If A(3) < E1 Then
      A(3) = E1
   Else
      E1 = A(3)
   End If
   For I = 1 To M
      Y1 = A(1) / (Sqr(2 * PI) * X(I))
      Y2 = (A(2) + A(1) * Log(A(3) - X(I))) ^ 2
      Y = Y1 * Exp(-0.5 * Y2)
      YTOT = YTOT * Y
```

```
   Next I
If YTOT > 0 Then FN = -Log(YTOT) Else FN = 1000000#

Case "S2"  'SPECIAL MDE OPTIMIZED
   If A(3) < E1 Then A(3) = E1 Else E1 = A(3)
   For I = 1 To M
      Z(I) = A(2) + A(1) * Log(A(3) - X(I))
   Next I
   Call TCFD(M, Z(), TC())
   If OC(1) = 0 Then
      Call OCFD(M, Z(), OC())
   End If
FN = DMAX(M, OC(), TC())

Case "B"  'BOUNDED MDE OPTIMIZED
   If A(3) = 0 Or A(4) = 0 Then
      A(3) = E1: A(4) = L1
   Else
      E1 = A(3): L1 = A(4)
   End If
   A(1) = Abs(A(1)) 'Prevents Delta < 0
   For I = 1 To M
      Z(I) = A(2) + A(1) * Log((X(I) - A(3)) / (A(3) + A(4) - X(I)))
   Next I
   Call TCFD(M, Z(), TC())
   If OC(1) = 0 Then
      Call OCFD(M, Z(), OC())
   End If
FN = DMAX(M, OC(), TC())

Case "U"  'UNBOUNDED MDE OPTIMIZED
   For I = 1 To M
      Z(I) = A(3) + A(1) * ArcSinh((X(I) - A(2)) / A(4))
   Next I
   Call TCFD(M, Z(), TC())
   If OC(1) = 0 Then
      Call OCFD(M, Z(), OC())
   End If
FN = DMAX(M, OC(), TC())

End Select
100 End Function
```

```
Function MIN3VAL(X1 As Double, X2 As Double, X3 As Double) As Double
'* THIS FUNCTION CALCULATES THE MINIMUM OF 3 VALUES.
If X1 <= X2 And X1 <= X3 Then
   MIN3VAL = X1
ElseIf X2 <= X1 And X2 <= X3 Then
   MIN3VAL = X2
Else
   MIN3VAL = X3
End If
End Function
```

```
Function MAX3VAL(X5 As Double, X6 As Double, X7 As Double) As Double
'* THIS FUNCTION CALCULATES THE MAXIMUM OF 3 VALUES.
If X5 >= X6 And X5 >= X7 Then
   MAX3VAL = X5
ElseIf X6 >= X5 And X6 >= X7 Then
   MAX3VAL = X6
Else
   MAX3VAL = X7
End If
End Function
```

```
Function MIN3POS(X1 As Double, X2 As Double, X3 As Double) As Integer
'* THIS FUNCTION DETERMINES THE POSITION OF THE MINIMUM VALUE.
If X1 <= X2 And X1 <= X3 Then
   MIN3POS = 1
ElseIf X2 <= X1 And X2 <= X3 Then
   MIN3POS = 2
Else
   MIN3POS = 3
End If
End Function
```

```
Function MAX3POS(X5 As Double, X6 As Double. X7 As Double) As Integer
'* THIS FUNCTION DETERMINES THE POSITION OF THE MAXIMUM VALUE.
If X5 >= X6 And X5 >= X7 Then
   MAX3POS = 5
ElseIf X6 >= X5 And X6 >= X7 Then
   MAX3POS = 6
Else
   MAX3POS = 7
End If
End Function
```

```
Function SIGN(NUM As Variant) As String
'* THIS FUNCTION DETERMINES THE TEXTUAL SIGN OF A VALUE.
If Sgn(NUM) < 0 Then
   SIGN = "-"
ElseIf Sgn(NUM) = 0 Then
   SIGN = " "
Else
   SIGN = "+"
End If
End Function
```

```
Function ArcSinh(X As Double) As Double
'* THIS FUNCTION CALCULATES THE INVERSE HYPERBOLIC SINE.
ArcSinh = Log(X + Sqr(X * X + 1))
End Function
```

```
Function Cosh(X As Double) As Double
'* THIS FUNCTION CALCULATES THE HYPERBOLIC COSINE.
Cosh = (Exp(X) + Exp(-X)) / 2
End Function
```

```
Function Sinh(X As Double) As Double
'* THIS FUNCTION CALCULATES THE HYPERBOLIC SINE.
Sinh = (Exp(X) - Exp(-X)) / 2
End Function
```

```
Function CB2(B1 As Double) As Double
'* THIS FUNCTION DETERMINES THE CALCULATED VALUE OF
'* B2, WHICH IS USED AGAINST THE OBSERVED VALUE OF B2
'* FROM THE DESCRIPTIVE STATISTICS.

Dim C As Double, W As Double

C = (2 / (2 + B1 + Sqr(B1 * (4 + B1)))) ^ (1 / 3)
W = (C - 1) ^ 2 / C + 1
CB2 = W ^ 4 + 2 * W ^ 3 + 3 * W ^ 2 - 3
End Function



Function CORRELXY(NV As Integer, DIST1 As String, X1() As Double, _
                  G1 As Double, D1 As Double, E1 As Double, L1 As Double, _
                  DIST2 As String, X2() As Double, G2 As Double, _
                  D2 As Double, E2 As Double, L2 As Double) As Double
'* THIS FUNCTION CALCULATES THE CORRELATION COEFFICIENT
'* BETWEEN THE Z-VALUES (STANDARD NORMAL TRANSFORMED)
'* OF TWO JOHNSON SYSTEM DISTRIBUTIONS.

Dim PM1 As Double, PM2 As Double, PM3 As Double, PM4 As Double
Dim PB1 As Double, PG1 As Double, PB2 As Double, PG2 As Double, B2C As Double
Dim Z1() As Double, Z2() As Double, I As Integer
ReDim Z1(1 To NV): ReDim Z2(1 To NV)

Call Stat1(X1(), NV, PM1, PM2, PM3, PM4, PB1, PG1, PB2, PG2)
Call STDZ(DIST1, NV, X1(), G1, D1, E1, L1, Z1())
Call Stat1(X2(), NV, PM1, PM2, PM3, PM4, PB1, PG1, PB2, PG2)
Call STDZ(DIST2, NV, X2(), G2, D2, E2, L2, Z2())
CORRELXY = Excel.Application.WorksheetFunction.Correl(Z1(), Z2())
End Function
```

```
'* THE VISUAL BASIC CODE LISTED AFTER THIS POINT REPRESENTS THE
'* CODE BEHIND FORM (CBF) OF THE SOFTWARE STARTING FORM IN
'* MICROSOFT ACCESS. IT IS LISTED HERE FOR REFERENCE ONLY.
Dim MySet As Recordset, MyDB As Database

Private Sub cmb_calculate_Click()
Dim Ifilepath As String, Ofilepath As String

Me!lbl_ispec.Visible = True
Me!lbl_ifile.Visible = True: Me!tbc_ifile.Visible = True: Me!cmb_ifile.Visible = True
Me!lbl_idrive.Visible = True: Me!tbc_idrive.Visible = True: Me!cmb_idrive.Visible =
True
Me!lbl_ipath.Visible = True: Me!tbc_ipath.Visible = True: Me!cmb_ipath.Visible = True
Me!lbl_ospec.Visible = True
Me!lbl_ofile.Visible = True: Me!tbc_ofile.Visible = True: Me!cmb_ofile.Visible = True
Me!lbl_odrive.Visible = True: Me!tbc_odrive.Visible = True: Me!cmb_odrive.Visible =
True
Me!lbl_opath.Visible = True: Me!tbc_opath.Visible = True: Me!cmb_opath.Visible =
True
Me!cmb_start.Visible = True
Set MyDB = CurrentDb
Set MySet = MyDB.OpenRecordset("File Specs", dbOpenTable)
Me!tbc_ifile.Value = MySet![Input Filename]
Me!tbc_idrive.Value = MySet![Input Drive]
If MySet![Input Filepath] = "New" Then
   Me!tbc_ipath.Value = CurDir("C")
   MySet.Edit: MySet![Input Filepath] = CurDir("C"): MySet.Update
Else
   Me!tbc_ipath.Value = MySet![Input Filepath]
End If
Me!tbc_ofile.Value = MySet![Output Filename]
Me!tbc_odrive.Value = MySet![Output Drive]
If MySet![Output Filepath] = "New" Then
   Me!tbc_opath.Value = CurDir("C")
   MySet.Edit: MySet![Output Filepath] = CurDir("C"): MySet.Update
Else
   Me!tbc_opath.Value = MySet![Output Filepath]
End If

If Right(Me!tbc_ipath.Value, 1) = "\" Then
   Ifilepath = Me!tbc_ipath.Value & Me!tbc_ifile.Value
Else
   Ifilepath = Me!tbc_ipath.Value & "\" & Me!tbc_ifile.Value
End If
If Right(Me!tbc_opath.Value, 1) = "\" Then
   Ofilepath = Me!tbc_opath.Value & Me!tbc_ofile.Value
```

```
Else
   Ofilepath = Me!tbc_opath.Value & "\" & Me!tbc_ofile.Value
End If
MySet.Edit: MySet![Input Totalpath] = Ifilepath: MySet.Update
MySet.Edit: MySet![Output Totalpath] = Ofilepath: MySet.Update
Me!cmb_ILink.Visible = True
Me!cmb_ILink.HyperlinkAddress = Ifilepath
Me!cmb_OLink.Visible = True
Me!cmb_OLink.HyperlinkAddress = Ofilepath
End Sub

Private Sub cmb_cancel_Click()
If Me!lbl_ospec.Visible = True Then
   Me!lbl_ispec.Visible = False
   Me!lbl_ifile.Visible = False: Me!tbc_ifile.Visible = False: Me!cmb_ifile.Visible =
False
   Me!lbl_idrive.Visible = False: Me!tbc_idrive.Visible = False: Me!cmb_idrive.Visible
= False
   Me!lbl_ipath.Visible = False: Me!tbc_ipath.Visible = False: Me!cmb_ipath.Visible =
False
   Me!lbl_ospec.Visible = False
   Me!lbl_ofile.Visible = False: Me!tbc_ofile.Visible = False: Me!cmb_ofile.Visible =
False
   Me!lbl_odrive.Visible = False: Me!tbc_odrive.Visible = False: Me!cmb_odrive.Visible
= False
   Me!lbl_opath.Visible = False: Me!tbc_opath.Visible = False: Me!cmb_opath.Visible =
False
   Me!cmb_start.Visible = False
Else
DoCmd.Close: DoCmd.Close
End If
End Sub

Private Sub cmb_generate_Click()
Me!lbl_ospec.Visible = True
Me!lbl_ofile.Visible = True: Me!tbc_ofile.Visible = True: Me!cmb_ofile.Visible = True
Me!lbl_odrive.Visible = True: Me!tbc_odrive.Visible = True: Me!cmb_odrive.Visible =
True
Me!lbl_opath.Visible = True: Me!tbc_opath.Visible = True: Me!cmb_opath.Visible =
True
Me!cmb_start.Visible = True
Me!cmb_ILink.Visible = False
Me!cmb_OLink.Visible = False
Set MyDB = CurrentDb
Set MySet = MyDB.OpenRecordset("File Specs", dbOpenTable)
Me!tbc_ofile.Value = MySet![Output GFilename]
```

```
Me!tbc_odrive.Value = MySet![Output GDrive]
If MySet![Output GFilepath] = "New" Then
   Me!tbc_opath.Value = CurDir("C")
   MySet.Edit: MySet![Output GFilepath] = CurDir("C"): MySet.Update
Else
   Me!tbc_opath.Value = MySet![Output GFilepath]
End If

End Sub

Private Sub cmb_idrive_Click()
Dim mess As String, title As String
Dim Ifilepath As String, Ofilepath As String
mess = "Enter the input file drive letter"
title = "CHANGE REQUEST"
10  Me!tbc_idrive.Value = InputBox(mess, title, MySet![Input Drive])
If Len(Me!tbc_idrive.Value) > 1 Or IsNumeric(Me!tbc_idrive.Value) = True Then
   mess = "Re-enter the input file drive letter (only)"
   title = "ERROR:  INVALID DRIVE LETTER"
   Me!tbc_idrive.Value = "INVALID DRIVE LETTER!"
   GoTo 10
Else
   MySet.Edit: MySet![Input Drive] = Me!tbc_idrive.Value: MySet.Update
   Me!tbc_ipath.Value = CurDir(Me!tbc_idrive.Value)
   MySet.Edit: MySet![Input Filepath] = Me!tbc_ipath.Value: MySet.Update
End If
If Right(Me!tbc_ipath.Value, 1) = "\" Then
   Ifilepath = Me!tbc_ipath.Value & Me!tbc_ifile.Value
Else
   Ifilepath = Me!tbc_ipath.Value & "\" & Me!tbc_ifile.Value
End If
If Right(Me!tbc_opath.Value, 1) = "\" Then
   Ofilepath = Me!tbc_opath.Value & Me!tbc_ofile.Value
Else
   Ofilepath = Me!tbc_opath.Value & "\" & Me!tbc_ofile.Value
End If
MySet.Edit: MySet![Input Totalpath] = Ifilepath: MySet.Update
MySet.Edit: MySet![Output Totalpath] = Ofilepath: MySet.Update
Me!cmb_ILink.HyperlinkAddress = Ifilepath
Me!cmb_OLink.HyperlinkAddress = Ofilepath
End Sub

Private Sub cmb_ifile_Click()
Dim mess As String, title As String
Dim Ifilepath As String, Ofilepath As String
mess = "Enter your input filename (with extension)"
```

```
title = "CHANGE REQUEST"
10  Me!tbc_ifile.Value = InputBox(mess, title, MySet![Input Filename])
If InStr(1, Me!tbc_ifile.Value, ".") < 1 Then
   mess = "Re-enter your input filename (with extension)"
   title = "ERROR:  INVALID FILENAME"
   Me!tbc_ifile.Value = "INVALID FILENAME!"
   GoTo 10
Else
   MySet.Edit: MySet![Input Filename] = Me!tbc_ifile.Value: MySet.Update
End If
If Right(Me!tbc_ipath.Value, 1) = "\" Then
   Ifilepath = Me!tbc_ipath.Value & Me!tbc_ifile.Value
Else
   Ifilepath = Me!tbc_ipath.Value & "\" & Me!tbc_ifile.Value
End If
If Right(Me!tbc_opath.Value, 1) = "\" Then
   Ofilepath = Me!tbc_opath.Value & Me!tbc_ofile.Value
Else
   Ofilepath = Me!tbc_opath.Value & "\" & Me!tbc_ofile.Value
End If
MySet.Edit: MySet![Input Totalpath] = Ifilepath: MySet.Update
MySet.Edit: MySet![Output Totalpath] = Ofilepath: MySet.Update
Me!cmb_ILink.HyperlinkAddress = Ifilepath
Me!cmb_OLink.HyperlinkAddress = Ofilepath
End Sub

Private Sub cmb_ipath_Click()
Dim mess As String, title As String
Dim Ifilepath As String, Ofilepath As String
mess = "Enter the full input filepath (without filename)"
title = "CHANGE REQUEST"
10  Me!tbc_ipath.Value = InputBox(mess, title, MySet![Input Filepath])
If InStr(1, Me!tbc_ipath.Value, Me!tbc_idrive.Value & ":\", vbTextCompare) < 1 Or _
     InStr(1, Me!tbc_ipath.Value, Me!tbc_ifile.Value, vbTextCompare) > 0 Or _
     InStr(1, Me!tbc_ipath.Value, Left(Me!tbc_ifile.Value, Len(Me!tbc_ifile.Value) - 4),
vbTextCompare) > 0 Then
   mess = "Re-enter the full input filepath (without filename)"
   title = "ERROR:  INVALID FILEPATH"
   Me!tbc_ipath.Value = "INVALID FILEPATH!"
   GoTo 10
Else
   MySet.Edit: MySet![Input Filepath] = Me!tbc_ipath.Value: MySet.Update
End If
If Right(Me!tbc_ipath.Value, 1) = "\" Then
   Ifilepath = Me!tbc_ipath.Value & Me!tbc_ifile.Value
Else
```

```
   Ifilepath = Me!tbc_ipath.Value & "\" & Me!tbc_ifile.Value
End If
If Right(Me!tbc_opath.Value, 1) = "\" Then
   Ofilepath = Me!tbc_opath.Value & Me!tbc_ofile.Value
Else
   Ofilepath = Me!tbc_opath.Value & "\" & Me!tbc_ofile.Value
End If
MySet.Edit: MySet![Input Totalpath] = Ifilepath: MySet.Update
MySet.Edit: MySet![Output Totalpath] = Ofilepath: MySet.Update
Me!cmb_ILink.HyperlinkAddress = Ifilepath
Me!cmb_OLink.HyperlinkAddress = Ofilepath
End Sub

Private Sub cmb_odrive_Click()
Dim mess As String, title As String
Dim Ifilepath As String, Ofilepath As String
mess = "Enter the output file drive letter"
title = "CHANGE REQUEST"
If Me!lbl_ispec.Visible = True Then
10     Me!tbc_odrive.Value = InputBox(mess, title, MySet![Output Drive])
   If Len(Me!tbc_odrive.Value) > 1 Or IsNumeric(Me!tbc_odrive.Value) = True Then
      mess = "Re-enter the output file drive letter (only)"
      title = "ERROR:  INVALID DRIVE LETTER"
      Me!tbc_odrive.Value = "INVALID DRIVE LETTER!"
      GoTo 10
   Else
      MySet.Edit: MySet![Output Drive] = Me!tbc_odrive.Value: MySet.Update
      Me!tbc_opath.Value = CurDir(Me!tbc_odrive.Value)
      MySet.Edit: MySet![Output Filepath] = Me!tbc_opath.Value: MySet.Update
   End If
   If Right(Me!tbc_ipath.Value, 1) = "\" Then
      Ifilepath = Me!tbc_ipath.Value & Me!tbc_ifile.Value
   Else
      Ifilepath = Me!tbc_ipath.Value & "\" & Me!tbc_ifile.Value
   End If
   If Right(Me!tbc_opath.Value, 1) = "\" Then
      Ofilepath = Me!tbc_opath.Value & Me!tbc_ofile.Value
   Else
      Ofilepath = Me!tbc_opath.Value & "\" & Me!tbc_ofile.Value
   End If
   MySet.Edit: MySet![Input Totalpath] = Ifilepath: MySet.Update
   MySet.Edit: MySet![Output Totalpath] = Ofilepath: MySet.Update
   Me!cmb_ILink.HyperlinkAddress = Ifilepath
   Me!cmb_OLink.HyperlinkAddress = Ofilepath
Else
20     Me!tbc_odrive.Value = InputBox(mess, title, MySet![Output GDrive])
```

```
  If Len(Me!tbc_odrive.Value) > 1 Or IsNumeric(Me!tbc_odrive.Value) = True Then
    mess = "Re-enter the output file drive letter (only)"
    title = "ERROR:  INVALID DRIVE LETTER"
    Me!tbc_odrive.Value = "INVALID DRIVE LETTER!"
    GoTo 20
  Else
    MySet.Edit: MySet![Output GDrive] = Me!tbc_odrive.Value: MySet.Update
    Me!tbc_opath.Value = CurDir(Me!tbc_odrive.Value)
    MySet.Edit: MySet![Output GFilepath] = Me!tbc_opath.Value: MySet.Update
  End If
End If
End Sub

Private Sub cmb_ofile_Click()
Dim mess As String, title As String
Dim Ifilepath As String, Ofilepath As String
mess = "Enter your output filename (with extension)"
title = "CHANGE REQUEST"
If Me!lbl_ispec.Visible = True Then
10    Me!tbc_ofile.Value = InputBox(mess, title, MySet![Output Filename])
  If InStr(1, Me!tbc_ofile.Value, ".") < 1 Then
    mess = "Re-enter your output filename (with extension)"
    title = "ERROR:  INVALID FILENAME"
    Me!tbc_ofile.Value = "INVALID FILENAME!"
    GoTo 10
  Else
    MySet.Edit: MySet![Output Filename] = Me!tbc_ofile.Value: MySet.Update
  End If
  If Right(Me!tbc_ipath.Value, 1) = "\" Then
    Ifilepath = Me!tbc_ipath.Value & Me!tbc_ifile.Value
  Else
    Ifilepath = Me!tbc_ipath.Value & "\" & Me!tbc_ifile.Value
  End If
  If Right(Me!tbc_opath.Value, 1) = "\" Then
    Ofilepath = Me!tbc_opath.Value & Me!tbc_ofile.Value
  Else
    Ofilepath = Me!tbc_opath.Value & "\" & Me!tbc_ofile.Value
  End If
  MySet.Edit: MySet![Input Totalpath] = Ifilepath: MySet.Update
  MySet.Edit: MySet![Output Totalpath] = Ofilepath: MySet.Update
  Me!cmb_ILink.HyperlinkAddress = Ifilepath
  Me!cmb_OLink.HyperlinkAddress = Ofilepath
Else
20    Me!tbc_ofile.Value = InputBox(mess, title, MySet![Output GFilename])
  If InStr(1, Me!tbc_ofile.Value, ".") < 1 Then
    mess = "Re-enter your output filename (with extension)"
```

```
      title = "ERROR:  INVALID FILENAME"
      Me!tbc_ofile.Value = "INVALID FILENAME!"
      GoTo 20
   Else
      MySet.Edit: MySet![Output GFilename] = Me!tbc_ofile.Value: MySet.Update
   End If
End If
End Sub

Private Sub cmb_opath_Click()
Dim mess As String, title As String
Dim Ifilepath As String, Ofilepath As String
mess = "Enter the full output filepath (without filename)"
title = "CHANGE REQUEST"
If Me!lbl_ispec.Visible = True Then
10     Me!tbc_opath.Value = InputBox(mess, title, MySet![Output Filepath])
   If InStr(1, Me!tbc_opath.Value, Me!tbc_odrive.Value & ":\", vbTextCompare) < 1 Or
_
         InStr(1, Me!tbc_opath.Value, Me!tbc_ofile.Value, vbTextCompare) > 0 Or _
         InStr(1, Me!tbc_opath.Value, Left(Me!tbc_ofile.Value, Len(Me!tbc_ofile.Value)
- 4), vbTextCompare) > 0 Then
      mess = "Re-enter the full output filepath (without filename)"
      title = "ERROR:  INVALID FILEPATH"
      Me!tbc_opath.Value = "INVALID FILEPATH!"
      GoTo 10
   Else
      MySet.Edit: MySet![Output Filepath] = Me!tbc_opath.Value: MySet.Update
   End If
   If Right(Me!tbc_ipath.Value, 1) = "\" Then
      Ifilepath = Me!tbc_ipath.Value & Me!tbc_ifile.Value
   Else
      Ifilepath = Me!tbc_ipath.Value & "\" & Me!tbc_ifile.Value
   End If
   If Right(Me!tbc_opath.Value, 1) = "\" Then
      Ofilepath = Me!tbc_opath.Value & Me!tbc_ofile.Value
   Else
      Ofilepath = Me!tbc_opath.Value & "\" & Me!tbc_ofile.Value
   End If
   MySet.Edit: MySet![Input Totalpath] = Ifilepath: MySet.Update
   MySet.Edit: MySet![Output Totalpath] = Ofilepath: MySet.Update
   Me!cmb_ILink.HyperlinkAddress = Ifilepath
   Me!cmb_OLink.HyperlinkAddress = Ofilepath
Else
20     Me!tbc_opath.Value = InputBox(mess, title, MySet![Output GFilepath])
   If InStr(1, Me!tbc_opath.Value, Me!tbc_odrive.Value & ":\", vbTextCompare) < 1 Or
_
```

```
        InStr(1, Me!tbc_opath.Value, Me!tbc_ofile.Value, vbTextCompare) > 0 Or _
        InStr(1, Me!tbc_opath.Value, Left(Me!tbc_ofile.Value, Len(Me!tbc_ofile.Value)
- 4), vbTextCompare) > 0 Then
      mess = "Re-enter the full output filepath (without filename)"
      title = "ERROR:  INVALID FILEPATH"
      Me!tbc_opath.Value = "INVALID FILEPATH!"
      GoTo 20
   Else
      MySet.Edit: MySet![Output GFilepath] = Me!tbc_opath.Value: MySet.Update
   End If
End If
End Sub

Private Sub cmb_start_Click()
Dim Ifilepath As String, Ofilepath As String
If Me!lbl_ispec.Visible = True Then
   If Right(Me!tbc_ipath.Value, 1) = "\" Then
      Ifilepath = Me!tbc_ipath.Value & Me!tbc_ifile.Value
   Else
      Ifilepath = Me!tbc_ipath.Value & "\" & Me!tbc_ifile.Value
   End If
   If Right(Me!tbc_opath.Value, 1) = "\" Then
      Ofilepath = Me!tbc_opath.Value & Me!tbc_ofile.Value
   Else
      Ofilepath = Me!tbc_opath.Value & "\" & Me!tbc_ofile.Value
   End If
   MySet.Edit: MySet![Input Totalpath] = Ifilepath: MySet.Update
   MySet.Edit: MySet![Output Totalpath] = Ofilepath: MySet.Update
   Call MAIN
Else
   If Right(Me!tbc_opath.Value, 1) = "\" Then
      Ofilepath = Me!tbc_opath.Value & Me!tbc_ofile.Value
   Else
      Ofilepath = Me!tbc_opath.Value & "\" & Me!tbc_ofile.Value
   End If
   MySet.Edit: MySet![Output GTotalpath] = Ofilepath: MySet.Update
   DoCmd.OpenForm "Generate Form", acNormal
End If
MySet.Close
Me!lbl_ispec.Visible = False
Me!lbl_ifile.Visible = False: Me!tbc_ifile.Visible = False: Me!cmb_ifile.Visible = False
Me!lbl_idrive.Visible = False: Me!tbc_idrive.Visible = False: Me!cmb_idrive.Visible =
False
Me!lbl_ipath.Visible = False: Me!tbc_ipath.Visible = False: Me!cmb_ipath.Visible =
False
Me!lbl_ospec.Visible = False
```

```
Me!lbl_ofile.Visible = False: Me!tbc_ofile.Visible = False: Me!cmb_ofile.Visible = False
Me!lbl_odrive.Visible = False: Me!tbc_odrive.Visible = False: Me!cmb_odrive.Visible =
False
Me!lbl_opath.Visible = False: Me!tbc_opath.Visible = False: Me!cmb_opath.Visible =
False
Me!cmb_cancel.SetFocus
Me!cmb_start.Visible = False
End Sub

Private Sub Form_Open(Cancel As Integer)
DoCmd.Maximize
Me!lbl_ispec.Visible = False
Me!lbl_ifile.Visible = False: Me!tbc_ifile.Visible = False: Me!cmb_ifile.Visible = False
Me!lbl_idrive.Visible = False: Me!tbc_idrive.Visible = False: Me!cmb_idrive.Visible =
False
Me!lbl_ipath.Visible = False: Me!tbc_ipath.Visible = False: Me!cmb_ipath.Visible =
False
Me!lbl_ospec.Visible = False
Me!lbl_ofile.Visible = False: Me!tbc_ofile.Visible = False: Me!cmb_ofile.Visible = False
Me!lbl_odrive.Visible = False: Me!tbc_odrive.Visible = False: Me!cmb_odrive.Visible =
False
Me!lbl_opath.Visible = False: Me!tbc_opath.Visible = False: Me!cmb_opath.Visible =
False
Me!cmb_start.Visible = False
Me!cmb_ILink.Visible = False: Me!cmb_OLink.Visible = False

End Sub
```

APPENDIX D

Multivariate Johnson Distribution Generator Program Code

```
Sub Generator()
'***********************************************************************
'* This Visual Basic Code Generates a Multivariate Sample of Specified Size from a
'* Specified Number of Johnson Systems of Specified Types.
'* Note that the Specified Correlation Coefficients Represent Correlation after
'* the Johnson System Transformations, and not of the Raw Samples.
'***********************************************************************
Dim NV As Integer, NS As Integer, I As Integer, J As Integer
Dim X() As Double, DIST() As String
Dim D() As Double, E() As Double, G() As Double, L() As Double
Dim P12 As Double, P24 As Double, P34 As Double
Dim P13 As Double, P23 As Double, P14 As Double
Dim LSL() As Double, USL() As Double
Dim Z1 As Double, Z21 As Double, Z312 As Double, Z4123 As Double
Dim EZ1 As Double, EZ21 As Double, EZ312 As Double, EZ4123 As Double
Dim VZ1 As Double, VZ21 As Double, VZ312 As Double, VZ4123 As Double
Dim A21 As Double, A312 As Double, B312 As Double, A4123 As Double
Dim B4123 As Double, C4123 As Double, Ofilepath As String
Dim MySet As Recordset, MyDB As Database

Const PI As Double = 3.14159265358979
Set MyDB = CurrentDb
Set MySet = MyDB.OpenRecordset("File Specs", dbOpenTable)
Ofilepath = MySet![Output GTotalpath]
MySet.Close

Open Ofilepath For Output As #1

NV = Forms![Generate Form]!tbc_NV.Value
NS = Forms![Generate Form]!tbc_NS.Value
Write #1, NV, NS

ReDim X(1 To NV)
ReDim DIST(1 To NV)
ReDim D(1 To NV)
ReDim E(1 To NV)
ReDim G(1 To NV)
ReDim L(1 To NV)
ReDim LSL(1 To NV)
ReDim USL(1 To NV)

DIST(1) = Forms![Generate Form]!cbx_Dist1.Value
E(1) = Forms![Generate Form]!tbc_E1.Value
L(1) = Forms![Generate Form]!tbc_L1.Value
D(1) = Forms![Generate Form]!tbc_D1.Value
G(1) = Forms![Generate Form]!tbc_G1.Value
```

```
LSL(1) = Forms![Generate Form]!tbc_LSL1.Value
USL(1) = Forms![Generate Form]!tbc_USL1.Value

If NV = 1 Then GoTo 5

DIST(2) = Forms![Generate Form]!cbx_Dist2.Value
E(2) = Forms![Generate Form]!tbc_E2.Value
L(2) = Forms![Generate Form]!tbc_L2.Value
D(2) = Forms![Generate Form]!tbc_D2.Value
G(2) = Forms![Generate Form]!tbc_G2.Value
P12 = Forms![Generate Form]!tbc_P12.Value
LSL(2) = Forms![Generate Form]!tbc_LSL2.Value
USL(2) = Forms![Generate Form]!tbc_USL2.Value

If NV = 2 Then GoTo 5

DIST(3) = Forms![Generate Form]!cbx_Dist3.Value
E(3) = Forms![Generate Form]!tbc_E3.Value
L(3) = Forms![Generate Form]!tbc_L3.Value
D(3) = Forms![Generate Form]!tbc_D3.Value
G(3) = Forms![Generate Form]!tbc_G3.Value
P13 = Forms![Generate Form]!tbc_P13.Value
P23 = Forms![Generate Form]!tbc_P23.Value
LSL(3) = Forms![Generate Form]!tbc_LSL3.Value
USL(3) = Forms![Generate Form]!tbc_USL3.Value

If NV = 3 Then GoTo 5

DIST(4) = Forms![Generate Form]!cbx_Dist4.Value
E(4) = Forms![Generate Form]!tbc_E4.Value
L(4) = Forms![Generate Form]!tbc_L4.Value
D(4) = Forms![Generate Form]!tbc_D4.Value
G(4) = Forms![Generate Form]!tbc_G4.Value
P14 = Forms![Generate Form]!tbc_P14.Value
P24 = Forms![Generate Form]!tbc_P24.Value
P34 = Forms![Generate Form]!tbc_P34.Value
LSL(4) = Forms![Generate Form]!tbc_LSL4.Value
USL(4) = Forms![Generate Form]!tbc_USL4.Value

5  If NV = 1 Then Write #1, LSL(1), USL(1)
If NV = 2 Then Write #1, LSL(1), USL(1), LSL(2), USL(2)
If NV = 3 Then Write #1, LSL(1), USL(1), LSL(2), USL(2), LSL(3), USL(3)
If NV = 4 Then Write #1, LSL(1), USL(1), LSL(2), USL(2), LSL(3), USL(3), LSL(4),
USL(4)

For I = 1 To NS
```

```
EZ1 = 0
VZ1 = 1
Z1 = EZ1 + VZ1 * Sqr(-2 * Log(Rnd())) * Cos(2 * PI * Rnd())
Select Case DIST(1)
   Case "N" 'Normal - (N*)
   X(1) = (Z1 - G(1)) / D(1)
   Case "L" 'LogNormal - (L*)
   X(1) = Exp((Z1 - G(1)) / D(1)) + E(1)
   Case "S" 'Special (S*)
   X(1) = E(1) - Exp((Z1 - G(1)) / D(1))
   Case "B" 'Bounded - (B*)
   X(1) = L(1) * (1 + Exp((G(1) - Z1) / D(1))) ^ (-1) + E(1)
   Case "U" 'UnBounded - (U*)
   X(1) = L(1) * Sinh((Z1 - G(1)) / D(1)) + E(1)
End Select

If NV = 1 Then GoTo 40
A21 = P12 / VZ1
EZ21 = (A21 * Z1)
VZ21 = 1 - (A21 * P12)
Z21 = EZ21 + Sqr(VZ21) * Sqr(-2 * Log(Rnd())) * Cos(2 * PI * Rnd())
Select Case DIST(2)
   Case "N" 'Normal - (NN*)
   X(2) = (Z21 - G(2)) / D(2)
   Case "L" 'LogNormal - (NL*)
   X(2) = Exp((Z21 - G(2)) / D(2)) + E(2)
   Case "S" 'Special - (NS*)
   X(2) = E(2) - Exp((Z21 - G(2)) / D(2))
   Case "B" 'Bounded - (NB*)
   X(2) = L(2) * (1 + Exp((G(2) - Z21) / D(2))) ^ (-1) + E(2)
   Case "U" 'UnBounded - (NU*)
   X(2) = L(2) * Sinh((Z21 - G(2)) / D(2)) + E(2)
End Select

If NV = 2 Then GoTo 40
A312 = (P13 - P12 * P23) / (VZ21 * VZ1)
B312 = (P23 - P12 * P13) / (VZ21 * VZ1)
EZ312 = (A312 * Z1) + (B312 * Z21)
VZ312 = 1 - ((A312 * P13) + (B312 * P23))
Z312 = EZ312 + Sqr(VZ312) * Sqr(-2 * Log(Rnd())) * Cos(2 * PI * Rnd())
Select Case DIST(3)
   Case "N" 'Normal - (NN*)
   X(3) = (Z312 - G(3)) / D(3)
   Case "L" 'LogNormal - (NL*)
   X(3) = Exp((Z312 - G(3)) / D(3)) + E(3)
   Case "S" 'Special - (NS*)
```

```
   X(3) = E(3) - Exp((Z312 - G(3)) / D(3))
   Case "B" 'Bounded - (NB*)
   X(3) = L(3) * (1 + Exp((G(3) - Z312) / D(3))) ^ (-1) + E(3)
   Case "U" 'UnBounded - (NU*)
   X(3) = L(3) * Sinh((Z312 - G(3)) / D(3)) + E(3)
End Select

If NV = 3 Then GoTo 40
A4123 = (P14 * (1 - P23 ^ 2) + P24 * (P13 * P23 - P12) + P34 * (P12 * P23 - P13)) / _
          (VZ312 * VZ21 * VZ1)
B4123 = (P14 * (P13 * P23 - P12) + P24 * (1 - P13 ^ 2) + P34 * (P12 * P13 - P23)) / _
          (VZ312 * VZ21 * VZ1)
C4123 = (P14 * (P12 * P23 - P13) + P24 * (P12 * P13 - P23) + P34 * (1 - P12 ^ 2)) / _
          (VZ312 * VZ21 * VZ1)
EZ4123 = (A4123 * Z1) + (B4123 * Z21) + (C4123 * Z312)
VZ4123 = 1 - ((A4123 * P14) + (B4123 * P24) + (C4123 * P34))
Z4123 = EZ4123 + Sqr(VZ4123) * Sqr(-2 * Log(Rnd())) * Cos(2 * PI * Rnd())
Select Case DIST(4)
   Case "N" 'Normal - (NN*)
   X(4) = (Z4123 - G(4)) / D(4)
   Case "L" 'LogNormal - (NL*)
   X(4) = Exp((Z4123 - G(4)) / D(4)) + E(4)
   Case "S" 'Special - (NS*)
   X(4) = E(4) - Exp((Z4123 - G(4)) / D(4))
   Case "B" 'Bounded - (NB*)
   X(4) = L(4) * (1 + Exp((G(4) - Z4123) / D(4))) ^ (-1) + E(4)
   Case "U" 'UnBounded - (NU*)
   X(4) = L(4) * Sinh((Z4123 - G(4)) / D(4)) + E(4)
End Select

40 'BEGINNING OF THE END
If NV = 1 Then Write #1, X(1)
If NV = 2 Then Write #1, X(1), X(2)
If NV = 3 Then Write #1, X(1), X(2), X(3)
If NV = 4 Then Write #1, X(1), X(2), X(3), X(4)
Next I
Close #1
MsgBox "DATA WRITTEN TO " & Ofilepath
End Sub
```

```
'* THE VISUAL BASIC CODE LISTED AFTER THIS POINT REPRESENTS THE
'* CODE BEHIND FORM (CBF) OF THE SOFTWARE'S JOHNSON SAMPLE
'* GENERATING FORM IN MICROSOFT ACCESS. IT IS LISTED HERE FOR
'* REFERENCE ONLY.

Private Sub cbx_Dist1_AfterUpdate()
If cbx_Dist1.Value = "L" Then
   tbc_L1.Value = 1: tbc_L1.Visible = False: lbl_L1.Visible = False
   tbc_E1.Value = Null: tbc_E1.Visible = True: lbl_E1.Visible = True
ElseIf cbx_Dist1.Value = "S" Then
   tbc_L1.Value = 1: tbc_L1.Visible = False: lbl_L1.Visible = False
   tbc_E1.Value = 0: tbc_E1.Visible = False: lbl_E1.Visible = False
ElseIf cbx_Dist1.Value = "N" Then
   tbc_L1.Value = 1: tbc_L1.Visible = False: lbl_L1.Visible = False
   tbc_E1.Value = 0: tbc_E1.Visible = False: lbl_E1.Visible = False
Else
   tbc_L1.Value = Null: tbc_L1.Visible = True: lbl_L1.Visible = True
   tbc_E1.Value = Null: tbc_E1.Visible = True: lbl_E1.Visible = True
   If cbx_Dist1.Value <> "B" And cbx_Dist1.Value <> "U" Then
      SendKeys "+{Tab}"
   End If
End If
End Sub

Private Sub cbx_Dist2_AfterUpdate()
If cbx_Dist2.Value = "L" Then
   tbc_L2.Value = 2: tbc_L2.Visible = False: lbl_L2.Visible = False
   tbc_E2.Value = Null: tbc_E2.Visible = True: lbl_E2.Visible = True
ElseIf cbx_Dist2.Value = "S" Then
   tbc_L2.Value = 2: tbc_L2.Visible = False: lbl_L2.Visible = False
   tbc_E2.Value = 0: tbc_E2.Visible = False: lbl_E2.Visible = False
ElseIf cbx_Dist2.Value = "N" Then
   tbc_L2.Value = 2: tbc_L2.Visible = False: lbl_L2.Visible = False
   tbc_E2.Value = 0: tbc_E2.Visible = False: lbl_E2.Visible = False
Else
   tbc_L2.Value = Null: tbc_L2.Visible = True: lbl_L2.Visible = True
   tbc_E2.Value = Null: tbc_E2.Visible = True: lbl_E2.Visible = True
   If cbx_Dist2.Value <> "B" And cbx_Dist2.Value <> "U" Then
      SendKeys "+{Tab}"
   End If
End If
End Sub

Private Sub cbx_Dist3_AfterUpdate()
If cbx_Dist3.Value = "L" Then
   tbc_L3.Value = 3: tbc_L3.Visible = False: lbl_L3.Visible = False
```

```
   tbc_E3.Value = Null: tbc_E3.Visible = True: lbl_E3.Visible = True
ElseIf cbx_Dist3.Value = "S" Then
   tbc_L3.Value = 3: tbc_L3.Visible = False: lbl_L3.Visible = False
   tbc_E3.Value = 0: tbc_E3.Visible = False: lbl_E3.Visible = False
ElseIf cbx_Dist3.Value = "N" Then
   tbc_L3.Value = 3: tbc_L3.Visible = False: lbl_L3.Visible = False
   tbc_E3.Value = 0: tbc_E3.Visible = False: lbl_E3.Visible = False
Else
   tbc_L3.Value = Null: tbc_L3.Visible = True: lbl_L3.Visible = True
   tbc_E3.Value = Null: tbc_E3.Visible = True: lbl_E3.Visible = True
   If cbx_Dist3.Value <> "B" And cbx_Dist3.Value <> "U" Then
      SendKeys "+{Tab}"
   End If
End If
End Sub

Private Sub cbx_Dist4_AfterUpdate()
If cbx_Dist4.Value = "L" Then
   tbc_L4.Value = 4: tbc_L4.Visible = False: lbl_L4.Visible = False
   tbc_E4.Value = Null: tbc_E4.Visible = True: lbl_E4.Visible = True
ElseIf cbx_Dist4.Value = "S" Then
   tbc_L4.Value = 4: tbc_L4.Visible = False: lbl_L4.Visible = False
   tbc_E4.Value = 0: tbc_E4.Visible = False: lbl_E4.Visible = False
ElseIf cbx_Dist4.Value = "N" Then
   tbc_L4.Value = 4: tbc_L4.Visible = False: lbl_L4.Visible = False
   tbc_E4.Value = 0: tbc_E4.Visible = False: lbl_E4.Visible = False
Else
   tbc_L4.Value = Null: tbc_L4.Visible = True: lbl_L4.Visible = True
   tbc_E4.Value = Null: tbc_E4.Visible = True: lbl_E4.Visible = True
   If cbx_Dist4.Value <> "B" And cbx_Dist4.Value <> "U" Then
      SendKeys "+{Tab}"
   End If
End If
End Sub

Private Sub cmb_cancel_Click()
DoCmd.Close
End Sub

Private Sub cmb_Generation_Click()
If IsNull(tbc_NV.Value) Then GoTo 10
If IsNull(tbc_NS.Value) Then GoTo 10
If IsNull(tbc_E1.Value) Then GoTo 10
If IsNull(tbc_L1.Value) Then GoTo 10
If IsNull(tbc_D1.Value) Then GoTo 10
If IsNull(tbc_G1.Value) Then GoTo 10
```

```
If IsNull(tbc_LSL1.Value) Then GoTo 10
If IsNull(tbc_USL1.Value) Then GoTo 10
If IsNull(cbx_Dist1.Value) Then GoTo 10
If tbc_NV.Value = 1 Then GoTo 5
If IsNull(tbc_E2.Value) Then GoTo 10
If IsNull(tbc_L2.Value) Then GoTo 10
If IsNull(tbc_D2.Value) Then GoTo 10
If IsNull(tbc_G2.Value) Then GoTo 10
If IsNull(tbc_P12.Value) Then GoTo 10
If IsNull(tbc_LSL2.Value) Then GoTo 10
If IsNull(tbc_USL2.Value) Then GoTo 10
If IsNull(cbx_Dist2.Value) Then GoTo 10
If tbc_NV.Value = 2 Then GoTo 5
If IsNull(tbc_E3.Value) Then GoTo 10
If IsNull(tbc_L3.Value) Then GoTo 10
If IsNull(tbc_D3.Value) Then GoTo 10
If IsNull(tbc_G3.Value) Then GoTo 10
If IsNull(tbc_P13.Value) Then GoTo 10
If IsNull(tbc_P23.Value) Then GoTo 10
If IsNull(tbc_LSL3.Value) Then GoTo 10
If IsNull(tbc_USL3.Value) Then GoTo 10
If IsNull(cbx_Dist3.Value) Then GoTo 10
If tbc_NV.Value = 3 Then GoTo 5
If IsNull(tbc_E4.Value) Then GoTo 10
If IsNull(tbc_L4.Value) Then GoTo 10
If IsNull(tbc_D4.Value) Then GoTo 10
If IsNull(tbc_G4.Value) Then GoTo 10
If IsNull(tbc_P14.Value) Then GoTo 10
If IsNull(tbc_P24.Value) Then GoTo 10
If IsNull(tbc_P34.Value) Then GoTo 10
If IsNull(tbc_LSL4.Value) Then GoTo 10
If IsNull(tbc_USL4.Value) Then GoTo 10
If IsNull(cbx_Dist4.Value) Then GoTo 10

5  Call Generator
DoCmd.Close
Exit Sub

10  MsgBox "Atleast one required field was found to be empty"
End Sub


Private Sub Form_Open(Cancel As Integer)
   Me!lbl_E2.Visible = False: Me!lbl_L2.Visible = False: Me!lbl_D2.Visible = False
   Me!lbl_G2.Visible = False: Me!lbl_dist2.Visible = False: Me!lbl_P12.Visible = False
   Me!lbl_LSL2.Visible = False: Me!lbl_USL2.Visible = False
```

```
    Me!tbc_E2.Visible = False: Me!tbc_L2.Visible = False: Me!tbc_D2.Visible = False
    Me!tbc_G2.Visible = False: Me!cbx_Dist2.Visible = False
    Me!tbc_P12.Visible = False
    Me!tbc_LSL2.Visible = False: Me!tbc_USL2.Visible = False
    Me!lbl_E3.Visible = False: Me!lbl_L3.Visible = False: Me!lbl_D3.Visible = False
    Me!lbl_G3.Visible = False: Me!lbl_Dist3.Visible = False: Me!lbl_P13.Visible = False
    Me!lbl_P23.Visible = False: Me!lbl_LSL3.Visible = False
    Me!lbl_USL3.Visible = False
    Me!tbc_E3.Visible = False: Me!tbc_L3.Visible = False: Me!tbc_D3.Visible = False
    Me!tbc_G3.Visible = False: Me!cbx_Dist3.Visible = False
    Me!tbc_P13.Visible = False
    Me!tbc_P23.Visible = False: Me!tbc_LSL3.Visible = False
    Me!tbc_USL3.Visible = False
    Me!lbl_E4.Visible = False: Me!lbl_L4.Visible = False: Me!lbl_D4.Visible = False
    Me!lbl_G4.Visible = False: Me!lbl_Dist4.Visible = False: Me!lbl_P14.Visible = False
    Me!lbl_P24.Visible = False: Me!lbl_P34.Visible = False
    Me!lbl_LSL4.Visible = False: Me!lbl_USL4.Visible = False
    Me!tbc_E4.Visible = False: Me!tbc_L4.Visible = False: Me!tbc_D4.Visible = False
    Me!tbc_G4.Visible = False: Me!cbx_Dist4.Visible = False
    Me!tbc_P14.Visible = False
    Me!tbc_P24.Visible = False: Me!tbc_P34.Visible = False
    Me!tbc_LSL4.Visible = False: Me!tbc_USL4.Visible = False
End Sub

Private Sub tbc_NV_AfterUpdate()
Dim NV As Integer
On Error GoTo Error_Handler
If IsNull(tbc_NV.Value) Then
    Me!lbl_E2.Visible = False: Me!lbl_L2.Visible = False: Me!lbl_D2.Visible = False
    Me!lbl_G2.Visible = False: Me!lbl_dist2.Visible = False: Me!lbl_P12.Visible = False
    Me!lbl_LSL2.Visible = False: Me!lbl_USL2.Visible = False
    Me!tbc_E2.Visible = False: Me!tbc_L2.Visible = False: Me!tbc_D2.Visible = False
    Me!tbc_G2.Visible = False: Me!cbx_Dist2.Visible = False
    Me!tbc_P12.Visible = False
    Me!tbc_LSL2.Visible = False: Me!tbc_USL2.Visible = False
    Me!lbl_E3.Visible = False: Me!lbl_L3.Visible = False: Me!lbl_D3.Visible = False
    Me!lbl_G3.Visible = False: Me!lbl_Dist3.Visible = False: Me!lbl_P13.Visible = False
    Me!lbl_P23.Visible = False: Me!lbl_LSL3.Visible = False
    Me!lbl_USL3.Visible = False
    Me!tbc_E3.Visible = False: Me!tbc_L3.Visible = False: Me!tbc_D3.Visible = False
    Me!tbc_G3.Visible = False: Me!cbx_Dist3.Visible = False
    Me!tbc_P13.Visible = False
    Me!tbc_P23.Visible = False: Me!tbc_LSL3.Visible = False
    Me!tbc_USL3.Visible = False
    Me!lbl_E4.Visible = False: Me!lbl_L4.Visible = False: Me!lbl_D4.Visible = False
    Me!lbl_G4.Visible = False: Me!lbl_Dist4.Visible = False: Me!lbl_P14.Visible = False
```

```
        Me!lbl_P24.Visible = False: Me!lbl_P34.Visible = False
        Me!lbl_LSL4.Visible = False: Me!lbl_USL4.Visible = False
        Me!tbc_E4.Visible = False: Me!tbc_L4.Visible = False: Me!tbc_D4.Visible = False
        Me!tbc_G4.Visible = False: Me!cbx_Dist4.Visible = False
        Me!tbc_P14.Visible = False
        Me!tbc_P24.Visible = False: Me!tbc_P34.Visible = False
        Me!tbc_LSL4.Visible = False
        Me!tbc_USL4.Visible = False
        SendKeys "+{Tab}"
        GoTo The_End
    End If
    NV = Me!tbc_NV.Value
    If NV = 1 Then
        Me!lbl_E2.Visible = False: Me!lbl_L2.Visible = False: Me!lbl_D2.Visible = False
        Me!lbl_G2.Visible = False: Me!lbl_dist2.Visible = False: Me!lbl_P12.Visible = False
        Me!lbl_LSL2.Visible = False: Me!lbl_USL2.Visible = False
        Me!tbc_E2.Visible = False: Me!tbc_L2.Visible = False: Me!tbc_D2.Visible = False
        Me!tbc_G2.Visible = False: Me!cbx_Dist2.Visible = False
        Me!tbc_P12.Visible = False
        Me!tbc_LSL2.Visible = False: Me!tbc_USL2.Visible = False
        Me!lbl_E3.Visible = False: Me!lbl_L3.Visible = False: Me!lbl_D3.Visible = False
        Me!lbl_G3.Visible = False: Me!lbl_Dist3.Visible = False: Me!lbl_P13.Visible = False
        Me!lbl_P23.Visible = False: Me!lbl_LSL3.Visible = False
        Me!lbl_USL3.Visible = False
        Me!tbc_E3.Visible = False: Me!tbc_L3.Visible = False: Me!tbc_D3.Visible = False
        Me!tbc_G3.Visible = False: Me!cbx_Dist3.Visible = False
        Me!tbc_P13.Visible = False
        Me!tbc_P23.Visible = False: Me!tbc_LSL3.Visible = False
        Me!tbc_USL3.Visible = False
        Me!lbl_E4.Visible = False: Me!lbl_L4.Visible = False: Me!lbl_D4.Visible = False
        Me!lbl_G4.Visible = False: Me!lbl_Dist4.Visible = False: Me!lbl_P14.Visible = False
        Me!lbl_P24.Visible = False: Me!lbl_P34.Visible = False
        Me!lbl_LSL4.Visible = False: Me!lbl_USL4.Visible = False
        Me!tbc_E4.Visible = False: Me!tbc_L4.Visible = False: Me!tbc_D4.Visible = False
        Me!tbc_G4.Visible = False: Me!cbx_Dist4.Visible = False
        Me!tbc_P14.Visible = False
        Me!tbc_P24.Visible = False: Me!tbc_P34.Visible = False
        Me!tbc_LSL4.Visible = False: Me!tbc_USL4.Visible = False
    ElseIf NV = 2 Then
        Me!lbl_E2.Visible = True: Me!lbl_L2.Visible = True: Me!lbl_D2.Visible = True
        Me!lbl_G2.Visible = True: Me!lbl_dist2.Visible = True: Me!lbl_P12.Visible = True
        Me!lbl_LSL2.Visible = True: Me!lbl_USL2.Visible = True
        Me!tbc_E2.Visible = True: Me!tbc_L2.Visible = True: Me!tbc_D2.Visible = True
        Me!tbc_G2.Visible = True: Me!cbx_Dist2.Visible = True: Me!tbc_P12.Visible = True
        Me!tbc_LSL2.Visible = True: Me!tbc_USL2.Visible = True
        Me!lbl_E3.Visible = False: Me!lbl_L3.Visible = False: Me!lbl_D3.Visible = False
```

```
        Me!lbl_G3.Visible = False: Me!lbl_Dist3.Visible = False: Me!lbl_P13.Visible = False
        Me!lbl_P23.Visible = False: Me!lbl_LSL3.Visible = False
        Me!lbl_USL3.Visible = False
        Me!tbc_E3.Visible = False: Me!tbc_L3.Visible = False: Me!tbc_D3.Visible = False
        Me!tbc_G3.Visible = False: Me!cbx_Dist3.Visible = False
        Me!tbc_P13.Visible = False
        Me!tbc_P23.Visible = False: Me!tbc_LSL3.Visible = False
        Me!tbc_USL3.Visible = False
        Me!lbl_E4.Visible = False: Me!lbl_L4.Visible = False: Me!lbl_D4.Visible = False
        Me!lbl_G4.Visible = False: Me!lbl_Dist4.Visible = False: Me!lbl_P14.Visible = False
        Me!lbl_P24.Visible = False: Me!lbl_P34.Visible = False
        Me!lbl_LSL4.Visible = False: Me!lbl_USL4.Visible = False
        Me!tbc_E4.Visible = False: Me!tbc_L4.Visible = False: Me!tbc_D4.Visible = False
        Me!tbc_G4.Visible = False: Me!cbx_Dist4.Visible = False
        Me!tbc_P14.Visible = False
        Me!tbc_P24.Visible = False: Me!tbc_P34.Visible = False
        Me!tbc_LSL4.Visible = False: Me!tbc_USL4.Visible = False
    ElseIf NV = 3 Then
        Me!lbl_E2.Visible = True: Me!lbl_L2.Visible = True: Me!lbl_D2.Visible = True
        Me!lbl_G2.Visible = True: Me!lbl_dist2.Visible = True: Me!lbl_P12.Visible = True
        Me!lbl_LSL2.Visible = True: Me!lbl_USL2.Visible = True
        Me!tbc_E2.Visible = True: Me!tbc_L2.Visible = True: Me!tbc_D2.Visible = True
        Me!tbc_G2.Visible = True: Me!cbx_Dist2.Visible = True: Me!tbc_P12.Visible = True
        Me!tbc_LSL2.Visible = True: Me!tbc_USL2.Visible = True
        Me!lbl_E3.Visible = True: Me!lbl_L3.Visible = True: Me!lbl_D3.Visible = True
        Me!lbl_G3.Visible = True: Me!lbl_Dist3.Visible = True: Me!lbl_P13.Visible = True
        Me!lbl_P23.Visible = True: Me!lbl_LSL3.Visible = True
        Me!lbl_USL3.Visible = True
        Me!tbc_E3.Visible = True: Me!tbc_L3.Visible = True: Me!tbc_D3.Visible = True
        Me!tbc_G3.Visible = True: Me!cbx_Dist3.Visible = True: Me!tbc_P13.Visible = True
        Me!tbc_P23.Visible = True: Me!tbc_LSL3.Visible = True
        Me!tbc_USL3.Visible = True
        Me!lbl_E4.Visible = False: Me!lbl_L4.Visible = False: Me!lbl_D4.Visible = False
        Me!lbl_G4.Visible = False: Me!lbl_Dist4.Visible = False: Me!lbl_P14.Visible = False
        Me!lbl_P24.Visible = False: Me!lbl_P34.Visible = False
        Me!lbl_LSL4.Visible = False: Me!lbl_USL4.Visible = False
        Me!tbc_E4.Visible = False: Me!tbc_L4.Visible = False: Me!tbc_D4.Visible = False
        Me!tbc_G4.Visible = False: Me!cbx_Dist4.Visible = False
        Me!tbc_P14.Visible = False
        Me!tbc_P24.Visible = False: Me!tbc_P34.Visible = False
        Me!tbc_LSL4.Visible = False: Me!tbc_USL4.Visible = False
    ElseIf NV = 4 Then
        Me!lbl_E2.Visible = True: Me!lbl_L2.Visible = True: Me!lbl_D2.Visible = True
        Me!lbl_G2.Visible = True: Me!lbl_dist2.Visible = True: Me!lbl_P12.Visible = True
        Me!lbl_LSL2.Visible = True: Me!lbl_USL2.Visible = True
        Me!tbc_E2.Visible = True: Me!tbc_L2.Visible = True: Me!tbc_D2.Visible = True
```

```
    Me!tbc_G2.Visible = True: Me!cbx_Dist2.Visible = True: Me!tbc_P12.Visible = True
    Me!tbc_LSL2.Visible = True: Me!tbc_USL2.Visible = True
    Me!lbl_E3.Visible = True: Me!lbl_L3.Visible = True: Me!lbl_D3.Visible = True
    Me!lbl_G3.Visible = True: Me!lbl_Dist3.Visible = True: Me!lbl_P13.Visible = True
    Me!lbl_P23.Visible = True: Me!lbl_LSL3.Visible = True
    Me!lbl_USL3.Visible = True
    Me!tbc_E3.Visible = True: Me!tbc_L3.Visible = True: Me!tbc_D3.Visible = True
    Me!tbc_G3.Visible = True: Me!cbx_Dist3.Visible = True: Me!tbc_P13.Visible = True
    Me!tbc_P23.Visible = True: Me!tbc_LSL3.Visible = True
    Me!tbc_USL3.Visible = True
    Me!lbl_E4.Visible = True: Me!lbl_L4.Visible = True: Me!lbl_D4.Visible = True
    Me!lbl_G4.Visible = True: Me!lbl_Dist4.Visible = True: Me!lbl_P14.Visible = True
    Me!lbl_P24.Visible = True: Me!lbl_P34.Visible = True: Me!lbl_LSL4.Visible = True
    Me!lbl_USL4.Visible = True
    Me!tbc_E4.Visible = True: Me!tbc_L4.Visible = True: Me!tbc_D4.Visible = True
    Me!tbc_G4.Visible = True: Me!cbx_Dist4.Visible = True: Me!tbc_P14.Visible = True
    Me!tbc_P24.Visible = True: Me!tbc_P34.Visible = True: Me!tbc_LSL4.Visible = True
    Me!tbc_USL4.Visible = True
Else
    Me!lbl_E2.Visible = False: Me!lbl_L2.Visible = False: Me!lbl_D2.Visible = False
    Me!lbl_G2.Visible = False: Me!lbl_dist2.Visible = False: Me!lbl_P12.Visible = False
    Me!lbl_LSL2.Visible = False: Me!lbl_USL2.Visible = False
    Me!tbc_E2.Visible = False: Me!tbc_L2.Visible = False: Me!tbc_D2.Visible = False
    Me!tbc_G2.Visible = False: Me!cbx_Dist2.Visible = False
    Me!tbc_P12.Visible = False
    Me!tbc_LSL2.Visible = False: Me!tbc_USL2.Visible = False
    Me!lbl_E3.Visible = False: Me!lbl_L3.Visible = False: Me!lbl_D3.Visible = False
    Me!lbl_G3.Visible = False: Me!lbl_Dist3.Visible = False: Me!lbl_P13.Visible = False
    Me!lbl_P23.Visible = False: Me!lbl_LSL3.Visible = False
    Me!lbl_USL3.Visible = False
    Me!tbc_E3.Visible = False: Me!tbc_L3.Visible = False: Me!tbc_D3.Visible = False
    Me!tbc_G3.Visible = False: Me!cbx_Dist3.Visible = False
    Me!tbc_P13.Visible = False
    Me!tbc_P23.Visible = False: Me!tbc_LSL3.Visible = False
    Me!tbc_USL3.Visible = False
    Me!lbl_E4.Visible = False: Me!lbl_L4.Visible = False: Me!lbl_D4.Visible = False
    Me!lbl_G4.Visible = False: Me!lbl_Dist4.Visible = False: Me!lbl_P14.Visible = False
    Me!lbl_P24.Visible = False: Me!lbl_P34.Visible = False
    Me!lbl_LSL4.Visible = False: Me!lbl_USL4.Visible = False
    Me!tbc_E4.Visible = False: Me!tbc_L4.Visible = False: Me!tbc_D4.Visible = False
    Me!tbc_G4.Visible = False: Me!cbx_Dist4.Visible = False
    Me!tbc_P14.Visible = False
    Me!tbc_P24.Visible = False: Me!tbc_P34.Visible = False
    Me!tbc_LSL4.Visible = False: Me!tbc_USL4.Visible = False
    MsgBox "Invalid Variable Number - Enter (1-4)"
    SendKeys "+{Tab}"
```

```
End If

The_End:
   Exit Sub

Error_Handler:
   MsgBox "Error " & Err.Number & ":  " & Err.Description
   Err.Clear
   SendKeys "+{Tab}"
   Resume The_End
   End Sub
```

APPENDIX E

Multivariate Conditional Johnson Median Equations

The median regression of x'_2 on x'_1 is studied because the means of the Johnson system equations are complex, when compared with the easy median equations. Since the median of a standard normal variable is zero, then the median of x'_2, given x'_1, satisfies the following equation:

$$\gamma_2 + \delta_2\, f_J(x'_2) = \rho_{12}\left[\gamma_1 + \delta_1\, f_I(x'_1)\right]$$

This equation can be rearranged as follows:

$$f_J(x'_2) = \frac{\rho_{12}\,\gamma_1 - \gamma_2}{\delta_2} + \left(\frac{\rho_{12}\,\delta_1}{\delta_2}\right) f_I(x'_1)$$

If we define θ and ϕ as follows:

$$\theta = exp\left(\frac{\rho_{12}\,\gamma_1 - \gamma_2}{\delta_2}\right) \qquad \text{and} \qquad \phi = \frac{\rho_{12}\,\delta_1}{\delta_2}$$

then the equation can be rewritten as follows:

$$f_J(x'_2) = ln(\theta) + \phi\, f_I(x'_1)$$

This form of the equation allows the derivation of the conditional bivariate Johnson equations. When the equations of the 16 different S_{IJ} distributions are considered, their derivations fall into four categories which are represented by the four Johnson system types which the S_J distribution can be. These categories would be S_{IN}, S_{IL}, S_{IB}, and S_{IU}. The logic behind the category selection will become clear as the derivations are presented.

The four distribution members of the S_{IN} category are S_{NN}, S_{LN}, S_{BN}, and S_{UN}. The derivation of the four median regression equations begins with the general equation, $f_J(x'_2) = ln(\theta) + \phi\, f_I(x'_1)$, where $f_J(x'_2) = f_N(x'_2) = x'_2$. The general form for the S_{IN} category regression equations would be:

S_{IN} category: $x'_2 = ln(\theta) + \phi f_I(x'_1)$

If we first consider the S_{NN} distribution where $f_I(x'_1) = f_N(x'_1) = x'_1$, the regression equation can be derived with a simple substitution:

S_{NN} distribution: $x'_2 = ln(\theta) + \phi x'_1$

The remaining three distributions in the S_{IN} category follow with their respective substitutions. For the S_{LN} distribution, the substitution is $f_I(x'_1) = f_L(x'_1) = ln(x'_1)$.

S_{LN} distribution: $x'_2 = ln(\theta) + \phi\, ln(x'_1)$

For the S_{BN} distribution, the substitution is $f_I(x'_1) = f_B(x'_1) = ln\left(\dfrac{x'_1}{1 - x'_1}\right)$.

S_{BN} distribution: $x'_2 = ln(\theta) + \phi\, ln\left(\dfrac{x'_1}{1 - x'_1}\right)$

For the S_{UN} distribution, the substitution is

$f_I(x'_1) = f_U(x'_1) = sinh^{-1}(x'_1) \equiv ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right)$.

S_{UN} distribution: $x'_2 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right)$

The four distribution members of the S_{IL} category are S_{NL}, S_{LL}, S_{BL}, and S_{UL}. The derivation of the four median regression equations begins with the general equation, $f_J(x'_2) = ln(\theta) + \phi f_I(x'_1)$, where $f_J(x'_2) = f_L(x'_2) = ln(x'_2)$. The general form for the S_{IL} category regression equations would be, $ln(x'_2) = ln(\theta) + \phi f_I(x'_1)$, which, by taking the exponent of each side, reduces to the following:

S_{IL} category: $x'_2 = \theta\, exp[\phi f_I(x'_1)]$

If we first consider the S_{NL} distribution where $f_I(x'_1) = f_N(x'_1) = x'_1$, the regression equation can be derived with a simple substitution:

S_{NL} distribution: $\quad x_2' = \theta \, exp\left[\phi \, x_1'\right]$

The remaining three distributions in the S_{IL} category follow with their respective substitutions with simplification. For the S_{LL} distribution, the substitution is $f_I(x_1') = f_L(x_1') = ln(x_1')$.

S_{LL} distribution: $\quad x_2' = \theta \left(x_1'\right)^{\phi}$

For the S_{BL} distribution, the substitution is $f_I(x_1') = f_B(x_1') = ln\left(\dfrac{x_1'}{1 - x_1'}\right)$.

S_{BL} distribution: $\quad x_2' = \theta \left(\dfrac{x_1'}{1 - x_1'}\right)^{\phi}$

For the S_{UL} distribution, the substitution is

$f_I(x_1') = f_U(x_1') = sinh^{-1}(x_1') \equiv ln\left(x_1' + \sqrt{x_1'^2 + 1}\right)$.

S_{UL} distribution: $\quad x_2' = \theta \left(x_1' + \sqrt{x_1'^2 + 1}\right)^{\phi}$

The four distribution members of the S_{IB} category are S_{NB}, S_{LB}, S_{BB}, and S_{UB}. The derivation of the four median regression equations begins with the general equation, $f_J(x_2') = ln(\theta) + \phi \, f_I(x_1')$, where $f_J(x_2') = f_B(x_2') = ln\left(\dfrac{x_2'}{1 - x_2'}\right)$. The substituted form for the S_{IB} category regression equations would be, $ln\left(\dfrac{x_2'}{1 - x_2'}\right) = ln(\theta) + \phi \, f_I(x_1')$, which, by taking the exponent of each side, reduces to, $\dfrac{x_2'}{1 - x_2'} = \theta \, exp\left[\phi \, f_I(x_1')\right]$. Some creative simplification will give us the general form of the equation as follows:

S_{IB} category: $\quad x_2' = \left\{1 + \theta^{-1} \, exp\left[-\phi \, f_I(x_1')\right]\right\}^{-1}$

If we first consider the S_{NB} distribution where $f_1(x_1') = f_N(x_1') = x_1'$, the regression equation can be derived with a simple substitution:

$$S_{NB} \text{ distribution:} \qquad x_2' = \left\{1 + \theta^{-1}\, exp\left[-\phi\, x_1'\right]\right\}^{-1}$$

The remaining three distributions in the S_{IB} category follow with their respective substitutions with simplification. For the S_{LB} distribution, the substitution is $f_1(x_1') = f_L(x_1') = ln(x_1')$.

$$S_{LB} \text{ distribution:} \qquad x_2' = \left[1 + \theta^{-1}\,(x_1')^{-\phi}\right]^{-1}$$

For the S_{BB} distribution, the substitution is $f_1(x_1') = f_B(x_1') = ln\left(\dfrac{x_1'}{1 - x_1'}\right)$.

$$S_{BB} \text{ distribution:} \qquad x_2' = \left[1 + \theta^{-1}\left(\frac{x_1'}{1 - x_1'}\right)^{-\phi}\right]^{-1}$$

For the S_{UB} distribution, the substitution is

$f_1(x_1') = f_U(x_1') = sinh^{-1}(x_1') \equiv ln\left(x_1' + \sqrt{x_1'^2 + 1}\right)$.

$$S_{UB} \text{ distribution:} \qquad x_2' = \left[1 + \theta^{-1}\left(x_1' + \sqrt{x_1'^2 + 1}\right)^{-\phi}\right]^{-1}$$

The four distribution members of the S_{IU} category are S_{NU}, S_{LU}, S_{BU}, and S_{UU}. The derivation of the four median regression equations begins with the general equation, $f_J(x_2') = ln(\theta) + \phi\, f_1(x_1')$, where $f_J(x_2') = f_U(x_2') = sinh^{-1}(x_2')$. The substituted form for the S_{IU} category regression equations would be, $sinh^{-1}(x_2') = ln(\theta) + \phi\, f_1(x_1')$, which, by taking the hyperbolic sine of each side, reduces to, $x_2' = sinh\left[ln(\theta) + \phi\, f_1(x_1')\right]$. From the definition of hyperbolic sine, $sinh(u) = \frac{1}{2}\left(e^{u} - e^{-u}\right)$, we can use substitution and simplification to give us the general form of the equation as follows:

$$S_{IU} \text{ category:} \quad x_2' = \frac{1}{2}\left\{\theta \, exp\left[\phi f_1(x_1')\right] - \theta^{-1} exp\left[-\phi f_1(x_1')\right]\right\}$$

If we first consider the S_{NU} distribution where $f_1(x_1') = f_N(x_1') = x_1'$, the regression equation can be derived with a simple substitution:

$$S_{NU} \text{ distribution:} \qquad x_2' = \frac{1}{2}\left\{\theta \, exp\left[\phi x_1'\right] - \theta^{-1} exp\left[-\phi x_1'\right]\right\}$$

The remaining three distributions in the S_{IU} category follow with their respective substitutions with simplification. For the S_{LU} distribution, the substitution is $f_1(x_1') = f_L(x_1') = ln(x_1')$.

$$S_{LU} \text{ distribution:} \qquad x_2' = \frac{1}{2}\left[\theta \, (x_1')^{\phi} - \theta^{-1} (x_1')^{-\phi}\right]$$

For the S_{BU} distribution, the substitution is $f_1(x_1') = f_B(x_1') = ln\left(\frac{x_1'}{1 - x_1'}\right)$.

$$S_{BU} \text{ distribution:} \qquad x_2' = \frac{1}{2}\left[\theta\left(\frac{x_1'}{1 - x_1'}\right)^{\phi} - \theta^{-1}\left(\frac{x_1'}{1 - x_1'}\right)^{-\phi}\right]$$

For the S_{UU} distribution, the substitution is

$f_1(x_1') = f_U(x_1') = sinh^{-1}(x_1') \equiv ln\left(x_1' + \sqrt{x_1'^2 + 1}\right)$.

$$S_{UU} \text{ distribution:} \qquad x_2' = \frac{1}{2}\left[\theta\left(x_1' + \sqrt{x_1'^2 + 1}\right)^{\phi} - \theta^{-1}\left(x_1' + \sqrt{x_1'^2 + 1}\right)^{-\phi}\right]$$

The median regression of x_3' on x_1' and x_2' is studied because the means of the Johnson system equations are complex, when compared with the easy median equations.

Since the median of a standard normal variable is zero, then the median of x'_3, given x'_1 and x'_2, satisfies the following equation:

$$\gamma_3 + \delta_3 f_K(x'_3) = \left(\frac{\rho_{13} - \rho_{12}\rho_{23}}{1-\rho_{12}^2}\right)\{\gamma_1 + \delta_1 f_I(x'_1)\} + \left(\frac{\rho_{23} - \rho_{12}\rho_{13}}{1-\rho_{12}^2}\right)\{\gamma_2 + \delta_2 f_J(x'_2)\}$$

This equation can be rearranged as follows:

$$f_K(x'_3) = \left[\frac{\left(\frac{\rho_{13} - \rho_{12}\rho_{23}}{1-\rho_{12}^2}\right)\gamma_1 - \left(\frac{\rho_{23} - \rho_{12}\rho_{13}}{1-\rho_{12}^2}\right)\gamma_2}{\delta_3}\right] + \left[\frac{\left(\frac{\rho_{13} - \rho_{12}\rho_{23}}{1-\rho_{12}^2}\right)\delta_1}{\delta_3}\right] f_I(x'_1) +$$
$$\left[\frac{\left(\frac{\rho_{23} - \rho_{12}\rho_{13}}{1-\rho_{12}^2}\right)\delta_2}{\delta_3}\right] f_J(x'_2)$$

If we define θ, ϕ, and α as follows:

$$\theta = exp\left[\frac{\left(\frac{\rho_{13} - \rho_{12}\rho_{23}}{1-\rho_{12}^2}\right)\gamma_1 - \left(\frac{\rho_{23} - \rho_{12}\rho_{13}}{1-\rho_{12}^2}\right)\gamma_2}{\delta_3}\right], \quad \phi = \left[\frac{\left(\frac{\rho_{13} - \rho_{12}\rho_{23}}{1-\rho_{12}^2}\right)\delta_1}{\delta_3}\right].$$

and $\alpha = \left[\frac{\left(\frac{\rho_{23} - \rho_{12}\rho_{13}}{1-\rho_{12}^2}\right)\delta_2}{\delta_3}\right]$

then the equation can be rewritten as follows:

$$f_K(x'_3) = ln(\theta) + \phi f_I(x'_1) + \alpha f_J(x'_2)$$

This form of the equation allows the derivation of the conditional trivariate Johnson equations. The equation derivations of the 64 different S_{IJK} distributions fall into four categories which are represented by the four Johnson system types which the S_K

distribution can be. These categories would be S_{IJN}, S_{IJL}, S_{IJB}, and S_{IJU}. The logic behind the category selection is the same as with the bivariate equations. The equations are derived in the same manner as the bivariate equations, presented earlier, and are listed below, following the quadrivariate discussion.

If we define θ, ϕ, α, and β using the same method as used with the bivariate and trivariate median regression equations, then the quadrivariate equation can be rewritten as follows:

$$f_M(x'_4) = ln(\theta) + \phi f_I(x'_1) + \alpha f_J(x'_2) + \beta f_K(x'_3)$$

This form of the equation allows the derivation of the conditional quadrivariate Johnson equations. The equation derivations of the 256 different S_{IJKM} distributions fall into four categories which are represented by the four Johnson system types which the S_M distribution can be. These categories would be S_{IJKN}, S_{IJKL}, S_{IJKB}, and S_{IJKU}. The logic behind the category selection is the same as with the bivariate and trivariate equations. The equations are derived in the same manner as the bivariate and trivariate equations, presented earlier, and are listed below, following the trivariate equations.

S_{IJN}: $x'_3 = ln(\theta) + \phi f_i(x'_1) + \alpha f_j(x'_2)$

S_{NNN}: $x'_3 = ln(\theta) + \phi x'_1 + \alpha x'_2$

S_{LNN}: $x'_3 = ln(\theta) + \phi ln(x'_1) + \alpha x'_2$

S_{BNN}: $x'_3 = ln(\theta) + \phi ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha x'_2$

S_{UNN}: $x'_3 = ln(\theta) + \phi ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha x'_2$

S_{NLN}: $x'_3 = ln(\theta) + \phi x'_1 + \alpha ln(x'_2)$

S_{LLN}: $x'_3 = ln(\theta) + \phi ln(x'_1) + \alpha ln(x'_2)$

S_{BLN}: $x'_3 = ln(\theta) + \phi ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha ln(x'_2)$

S_{ULN}: $x'_3 = ln(\theta) + \phi ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha ln(x'_2)$

S_{NBN}: $x'_3 = ln(\theta) + \phi x'_1 + \alpha ln\left(\frac{x'_2}{1-x'_2}\right)$

S_{LBN}: $x'_3 = ln(\theta) + \phi ln(x'_1) + \alpha ln\left(\frac{x'_2}{1-x'_2}\right)$

S_{BBN}: $x'_3 = ln(\theta) + \phi ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha ln\left(\frac{x'_2}{1-x'_2}\right)$

S_{UBN}: $x'_3 = ln(\theta) + \phi ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha ln\left(\frac{x'_2}{1-x'_2}\right)$

S_{NUN}: $x'_3 = ln(\theta) + \phi x'_1 + \alpha ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right)$

S_{LUN}: $x'_3 = ln(\theta) + \phi ln(x'_1) + \alpha ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right)$

S_{BUN}: $x'_3 = ln(\theta) + \phi ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right)$

S_{UUN}: $x'_3 = ln(\theta) + \phi ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right)$

S_{IJL}: $x'_3 = \theta\, exp[\phi f_i(x'_1)]\, exp[\alpha f_j(x'_2)]$

S_{NNL}: $x'_3 = \theta\, exp[\phi x'_1]\, exp[\alpha x'_2]$

S_{LNL}: $x'_3 = \theta\, (x'_1)^{\phi}\, exp[\alpha x'_2]$

S_{BNL}: $$x'_3 = \theta \left(\frac{x'_1}{1 - x'_1} \right)^{\phi} exp\left[\alpha\, x'_2\right]$$

S_{UNL}: $$x'_3 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1} \right)^{\phi} exp\left[\alpha\, x'_2\right]$$

S_{NLL}: $$x'_3 = \theta\, exp\left[\phi\, x'_1\right] \left(x'_2 \right)^{\alpha}$$

S_{LLL}: $$x'_3 = \theta \left(x'_1 \right)^{\phi} \left(x'_2 \right)^{\alpha}$$

S_{BLL}: $$x'_3 = \theta \left(\frac{x'_1}{1 - x'_1} \right)^{\phi} \left(x'_2 \right)^{\alpha}$$

S_{ULL}: $$x'_3 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1} \right)^{\phi} \left(x'_2 \right)^{\alpha}$$

S_{NBL}: $$x'_3 = \theta\, exp\left[\phi\, x'_1\right] \left(\frac{x'_2}{1 - x'_2} \right)^{\alpha}$$

S_{LBL}: $$x'_3 = \theta \left(x'_1 \right)^{\phi} \left(\frac{x'_2}{1 - x'_2} \right)^{\alpha}$$

S_{BBL}: $$x'_3 = \theta \left(\frac{x'_1}{1 - x'_1} \right)^{\phi} \left(\frac{x'_2}{1 - x'_2} \right)^{\alpha}$$

S_{UBL}: $$x'_3 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1} \right)^{\phi} \left(\frac{x'_2}{1 - x'_2} \right)^{\alpha}$$

S_{NUL}: $$x'_3 = \theta\, exp\left[\phi\, x'_1\right] \left(x'_2 + \sqrt{x'^2_2 + 1} \right)^{\alpha}$$

S_{LUL}: $$x'_3 = \theta \left(x'_1 \right)^{\phi} \left(x'_2 + \sqrt{x'^2_2 + 1} \right)^{\alpha}$$

S_{BUL}: $$x'_3 = \theta \left(\frac{x'_1}{1 - x'_1} \right)^{\phi} \left(x'_2 + \sqrt{x'^2_2 + 1} \right)^{\alpha}$$

S_{UUL}: $$x'_3 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1} \right)^{\phi} \left(x'_2 + \sqrt{x'^2_2 + 1} \right)^{\alpha}$$

S_{IJB}: $$x'_3 = \left\{ 1 + \theta^{-1}\, exp\left[-\phi\, f_I\left(x'_1\right)\right] exp\left[-\alpha\, f_J\left(x'_2\right)\right] \right\}^{-1}$$

S_{NNB}: $$x'_3 = \left\{ 1 + \theta^{-1}\, exp\left[-\phi\, x'_1\right] exp\left[-\alpha\, x'_2\right] \right\}^{-1}$$

S_{LNB}: $$x'_3 = \left\{ 1 + \theta^{-1} \left(x'_1 \right)^{-\phi} exp\left[-\alpha\, x'_2\right] \right\}^{-1}$$

S_{BNB}: $$x'_3 = \left\{ 1 + \theta^{-1} \left(\frac{x'_1}{1 - x'_1} \right)^{-\phi} exp\left[-\alpha\, x'_2\right] \right\}^{-1}$$

S_{UNB}: $$x'_3 = \left\{ 1 + \theta^{-1} \left(x'_1 + \sqrt{x'^2_1 + 1} \right)^{-\phi} exp\left[-\alpha\, x'_2\right] \right\}^{-1}$$

S_{NLB}: $$x'_3 = \left\{ 1 + \theta^{-1}\, exp\left[-\phi\, x'_1\right] \left(x'_2 \right)^{-\alpha} \right\}^{-1}$$

S_{LLB}: $$x'_3 = \left\{1 + \theta^{-1} \left(x'_1\right)^{-\phi} \left(x'_2\right)^{-\alpha}\right\}^{-1}$$

S_{BLB}: $$x'_3 = \left\{1 + \theta^{-1} \left(\frac{x'_1}{1 - x'_1}\right)^{-\phi} \left(x'_2\right)^{-\alpha}\right\}^{-1}$$

S_{ULB}: $$x'_3 = \left\{1 + \theta^{-1} \left(x'_1 + \sqrt{x_1'^2 + 1}\right)^{-\phi} \left(x'_2\right)^{-\alpha}\right\}^{-1}$$

S_{NBB}: $$x'_3 = \left\{1 + \theta^{-1}\, exp\left[-\phi\, x'_1\right] \left(\frac{x'_2}{1 - x'_2}\right)^{-\alpha}\right\}^{-1}$$

S_{LBB}: $$x'_3 = \left\{1 + \theta^{-1} \left(x'_1\right)^{-\phi} \left(\frac{x'_2}{1 - x'_2}\right)^{-\alpha}\right\}^{-1}$$

S_{BBB}: $$x'_3 = \left\{1 + \theta^{-1} \left(\frac{x'_1}{1 - x'_1}\right)^{-\phi} \left(\frac{x'_2}{1 - x'_2}\right)^{-\alpha}\right\}^{-1}$$

S_{UBB}: $$x'_3 = \left\{1 + \theta^{-1} \left(x'_1 + \sqrt{x_1'^2 + 1}\right)^{-\phi} \left(\frac{x'_2}{1 - x'_2}\right)^{-\alpha}\right\}^{-1}$$

S_{NUB}: $$x'_3 = \left\{1 + \theta^{-1}\, exp\left[-\phi\, x'_1\right] \left(x'_2 + \sqrt{x_2'^2 + 1}\right)^{-\alpha}\right\}^{-1}$$

S_{LUB}: $$x'_3 = \left\{1 + \theta^{-1} \left(x'_1\right)^{-\phi} \left(x'_2 + \sqrt{x_2'^2 + 1}\right)^{-\alpha}\right\}^{-1}$$

S_{BUB}: $$x'_3 = \left\{1 + \theta^{-1} \left(\frac{x'_1}{1 - x'_1}\right)^{-\phi} \left(x'_2 + \sqrt{x_2'^2 + 1}\right)^{-\alpha}\right\}^{-1}$$

S_{UUB}: $$x'_3 = \left\{1 + \theta^{-1} \left(x'_1 + \sqrt{x_1'^2 + 1}\right)^{-\phi} \left(x'_2 + \sqrt{x_2'^2 + 1}\right)^{-\alpha}\right\}^{-1}$$

S_{IJU}: $$x'_3 = \frac{1}{2}\left\{\theta\, exp\left[\phi\, f_1(x'_1)\right] exp\left[\alpha\, f_1(x'_2)\right] - \theta^{-1}\, exp\left[-\phi\, f_1(x'_1)\right] exp\left[-\alpha\, f_1(x'_2)\right]\right\}$$

S_{NNU}: $$x'_3 = \frac{1}{2}\left\{\theta\, exp\left[\phi\, x'_1\right] exp\left[\alpha\, x'_2\right] - \theta^{-1}\, exp\left[-\phi\, x'_1\right] exp\left[-\alpha\, x'_2\right]\right\}$$

S_{LNU}: $$x'_3 = \frac{1}{2}\left\{\theta \left(x'_1\right)^{\phi} exp\left[\alpha\, x'_2\right] - \theta^{-1} \left(x'_1\right)^{-\phi} exp\left[-\alpha\, x'_2\right]\right\}$$

S_{BNU}: $$x'_2 = \frac{1}{2}\left\{\theta \left(\frac{x'_1}{1 - x'_1}\right)^{\phi} exp\left[\alpha\, x'_2\right] - \theta^{-1} \left(\frac{x'_1}{1 - x'_1}\right)^{-\phi} exp\left[-\alpha\, x'_2\right]\right\}$$

S_{UNU}: $$x'_3 = \frac{1}{2}\left\{\theta \left(x'_1 + \sqrt{x_1'^2 + 1}\right)^{\phi} exp\left[\alpha\, x'_2\right] - \theta^{-1} \left(x'_1 + \sqrt{x_1'^2 + 1}\right)^{-\phi} exp\left[-\alpha\, x'_2\right]\right\}$$

S_{NLU}: $$x_3' = \frac{1}{2}\left\{\theta\, exp\left[\phi\, x_1'\right]\left(x_2'\right)^{\alpha} - \theta^{-1}\, exp\left[-\phi\, x_1'\right]\left(x_2'\right)^{-\alpha}\right\}$$

S_{LLU}: $$x_3' = \frac{1}{2}\left\{\theta\left(x_1'\right)^{\phi}\left(x_2'\right)^{\alpha} - \theta^{-1}\left(x_1'\right)^{-\phi}\left(x_2'\right)^{-\alpha}\right\}$$

S_{BLU}: $$x_3' = \frac{1}{2}\left\{\theta\left(\frac{x_1'}{1-x_1'}\right)^{\phi}\left(x_2'\right)^{\alpha} - \theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(x_2'\right)^{-\alpha}\right\}$$

S_{ULU}: $$x_3' = \frac{1}{2}\left\{\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}\left(x_2'\right)^{\alpha} - \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}\left(x_2'\right)^{-\alpha}\right\}$$

S_{NBU}: $$x_3' = \frac{1}{2}\left\{\theta\, exp\left[\phi\, x_1'\right]\left(\frac{x_2'}{1-x_2'}\right)^{\alpha} - \theta^{-1}\, exp\left[-\phi\, x_1'\right]\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\right\}$$

S_{LBU}: $$x_3' = \frac{1}{2}\left\{\theta\left(x_1'\right)^{\phi}\left(\frac{x_2'}{1-x_2'}\right)^{\alpha} - \theta^{-1}\left(x_1'\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\right\}$$

S_{BBU}: $$x_3' = \frac{1}{2}\left\{\theta\left(\frac{x_1'}{1-x_1'}\right)^{\phi}\left(\frac{x_2'}{1-x_2'}\right)^{\alpha} - \theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\right\}$$

S_{UBU}: $$x_3' = \frac{1}{2}\left\{\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}\left(\frac{x_2'}{1-x_2'}\right)^{\alpha} - \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\right\}$$

S_{NUU}: $$x_3' = \frac{1}{2}\left\{\theta\, exp\left[\phi\, x_1'\right]\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha} - \theta^{-1}\, exp\left[-\phi\, x_1'\right]\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}\right\}$$

S_{LUU}: $$x_3' = \frac{1}{2}\left\{\theta\left(x_1'\right)^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha} - \theta^{-1}\left(x_1'\right)^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}\right\}$$

S_{BUU}: $$x_3' = \frac{1}{2}\left\{\theta\left(\frac{x_1'}{1-x_1'}\right)^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha} - \theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}\right\}$$

S_{UUU}: $$x_3' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha} - \\ \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}\end{array}\right\}$$

Quadrivariate Equations

$$S_{IJNN}:\ x'_4 = ln(\theta) + \phi\, f_I(x'_1) + \alpha\, f_J(x'_2) + \beta\, x'_3$$

$$S_{NNNN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, x'_2 + \beta\, x'_3$$
$$S_{LNNN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, x'_2 + \beta\, x'_3$$
$$S_{BNNN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, x'_2 + \beta\, x'_3$$
$$S_{UNNN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha\, x'_2 + \beta\, x'_3$$
$$S_{NLNN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln(x'_2) + \beta\, x'_3$$
$$S_{LLNN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln(x'_2) + \beta\, x'_3$$
$$S_{BLNN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, ln(x'_2) + \beta\, x'_3$$
$$S_{ULNN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha\, ln(x'_2) + \beta\, x'_3$$
$$S_{NBNN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, x'_3$$
$$S_{LBNN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, x'_3$$
$$S_{BBNN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, x'_3$$
$$S_{UBNN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, x'_3$$
$$S_{NUNN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, x'_3$$
$$S_{LUNN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, x'_3$$
$$S_{BUNN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, x'_3$$
$$S_{UUNN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, x'_3$$

$$S_{IJLN}:\ x'_4 = ln(\theta) + \phi\, f_I(x'_1) + \alpha\, f_J(x'_2) + \beta\, ln(x'_3)$$

$$S_{NNLN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, x'_2 + \beta\, ln(x'_3)$$
$$S_{LNLN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, x'_2 + \beta\, ln(x'_3)$$
$$S_{BNLN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, x'_2 + \beta\, ln(x'_3)$$

$$S_{UNLN}:\; x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x_1'^2 + 1}\right) + \alpha\, x'_2 + \beta\, ln(x'_3)$$

$$S_{NLLN}:\; x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln(x'_2) + \beta\, ln(x'_3)$$

$$S_{LLLN}:\; x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln(x'_2) + \beta\, ln(x'_3)$$

$$S_{BLLN}:\; x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1 - x'_1}\right) + \alpha\, ln(x'_2) + \beta\, ln(x'_3)$$

$$S_{ULLN}:\; x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x_1'^2 + 1}\right) + \alpha\, ln(x'_2) + \beta\, ln(x'_3)$$

$$S_{NBLN}:\; x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln\left(\frac{x'_2}{1 - x'_2}\right) + \beta\, ln(x'_3)$$

$$S_{LBLN}:\; x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln\left(\frac{x'_2}{1 - x'_2}\right) + \beta\, ln(x'_3)$$

$$S_{BBLN}:\; x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1 - x'_1}\right) + \alpha\, ln\left(\frac{x'_2}{1 - x'_2}\right) + \beta\, ln(x'_3)$$

$$S_{UBLN}:\; x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x_1'^2 + 1}\right) + \alpha\, ln\left(\frac{x'_2}{1 - x'_2}\right) + \beta\, ln(x'_3)$$

$$S_{NULN}:\; x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln\left(x'_2 + \sqrt{x_2'^2 + 1}\right) + \beta\, ln(x'_3)$$

$$S_{LULN}:\; x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln\left(x'_2 + \sqrt{x_2'^2 + 1}\right) + \beta\, ln(x'_3)$$

$$S_{BULN}:\; x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1 - x'_1}\right) + \alpha\, ln\left(x'_2 + \sqrt{x_2'^2 + 1}\right) + \beta\, ln(x'_3)$$

$$S_{UULN}:\; x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x_1'^2 + 1}\right) + \alpha\, ln\left(x'_2 + \sqrt{x_2'^2 + 1}\right) + \beta\, ln(x'_3)$$

$$S_{IJBN}:\; x'_4 = ln(\theta) + \phi\, f_i(x'_1) + \alpha\, f_j(x'_2) + \beta\, ln\left(\frac{x'_3}{1 - x'_3}\right)$$

$$S_{NNBN}:\; x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, x'_2 + \beta\, ln\left(\frac{x'_3}{1 - x'_3}\right)$$

$$S_{LNBN}:\; x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, x'_2 + \beta\, ln\left(\frac{x'_3}{1 - x'_3}\right)$$

$$S_{BNBN}:\; x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1 - x'_1}\right) + \alpha\, x'_2 + \beta\, ln\left(\frac{x'_3}{1 - x'_3}\right)$$

$$S_{UNBN}:\; x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x_1'^2 + 1}\right) + \alpha\, x'_2 + \beta\, ln\left(\frac{x'_3}{1 - x'_3}\right)$$

$$S_{NLBN}:\; x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln(x'_2) + \beta\, ln\left(\frac{x'_3}{1 - x'_3}\right)$$

$$S_{LLBN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln(x'_2) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{BLBN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, ln(x'_2) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{ULBN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha\, ln(x'_2) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{NBBN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{LBBN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{BBBN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{UBBN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{NUBN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{LUBN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{BUBN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{UUBN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, ln\left(\frac{x'_3}{1-x'_3}\right)$$

$$S_{IJUN}:\ x'_4 = ln(\theta) + \phi\, f_1(x'_1) + \alpha\, f_2(x'_2) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{NNUN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, x'_2 + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{LNUN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, x'_2 + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{BNUN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, x'_2 + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{UNUN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha\, x'_2 + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{NLUN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln(x'_2) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{LLUN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln(x'_2) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{BLUN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, ln(x'_2) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{ULUN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha\, ln(x'_2) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{NBUN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{LBUN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{BBUN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{UBUN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2 + 1}\right) + \alpha\, ln\left(\frac{x'_2}{1-x'_2}\right) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{NUUN}:\ x'_4 = ln(\theta) + \phi\, x'_1 + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{LUUN}:\ x'_4 = ln(\theta) + \phi\, ln(x'_1) + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{BUUN}:\ x'_4 = ln(\theta) + \phi\, ln\left(\frac{x'_1}{1-x'_1}\right) + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{UUUN}:\ x'_4 = ln(\theta) + \phi\, ln\left(x'_1 + \sqrt{x'^2_1 + 1}\right) + \alpha\, ln\left(x'_2 + \sqrt{x'^2_2 + 1}\right) + \beta\, ln\left(x'_3 + \sqrt{x'^2_3 + 1}\right)$$

$$S_{IJNL}:\ x'_4 = \theta\, exp\left[\phi\, f_1(x'_1)\right] exp\left[\alpha\, f_1(x'_2)\right] exp\left[\beta\, x'_3\right]$$

$$S_{NNNL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right] exp\left[\alpha\, x'_2\right] exp\left[\beta\, x'_3\right]$$

$$S_{LNNL}:\ x'_4 = \theta\, (x'_1)^{\phi}\ exp\left[\alpha\, x'_2\right] exp\left[\beta\, x'_3\right]$$

$$S_{BNNL}:\ x'_4 = \theta\left(\frac{x'_1}{1-x'_1}\right)^{\phi} exp\left[\alpha\, x'_2\right] exp\left[\beta\, x'_3\right]$$

$$S_{UNNL}:\ x'_4 = \theta\left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{\phi} exp\left[\alpha\, x'_2\right] exp\left[\beta\, x'_3\right]$$

$$S_{NLNL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right] (x'_2)^{\alpha}\ exp\left[\beta\, x'_3\right]$$

$$S_{LLNL}:\ x'_4 = \theta\, (x'_1)^{\phi}\ (x'_2)^{\alpha}\ exp\left[\beta\, x'_3\right]$$

$$S_{BLNL}:\ x'_4 = \theta\left(\frac{x'_1}{1-x'_1}\right)^{\phi} (x'_2)^{\alpha}\ exp\left[\beta\, x'_3\right]$$

$$S_{ULNL}:\ x'_4 = \theta\left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{\phi} (x'_2)^{\alpha}\ exp\left[\beta\, x'_3\right]$$

$$S_{NBNL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right]\left(\frac{x'_2}{1-x'_2}\right)^{\alpha} exp\left[\beta\, x'_3\right]$$

$$S_{LBNL}:\ x'_4 = \theta \left(x'_1\right)^{\phi} \left(\frac{x'_2}{1-x'_2}\right)^{\alpha} exp\left[\beta x'_3\right]$$

$$S_{BBNL}:\ x'_4 = \theta \left(\frac{x'_1}{1-x'_1}\right)^{\phi} \left(\frac{x'_2}{1-x'_2}\right)^{\alpha} exp\left[\beta x'_3\right]$$

$$S_{UBNL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x_1'^2+1}\right)^{\phi} \left(\frac{x'_2}{1-x'_2}\right)^{\alpha} exp\left[\beta x'_3\right]$$

$$S_{NUNL}:\ x'_4 = \theta\, exp\left[\phi x'_1\right]\left(x'_2 + \sqrt{x_2'^2+1}\right)^{\alpha} exp\left[\beta x'_3\right]$$

$$S_{LUNL}:\ x'_4 = \theta \left(x'_1\right)^{\phi} \left(x'_2 + \sqrt{x_2'^2+1}\right)^{\alpha} exp\left[\beta x'_3\right]$$

$$S_{BUNL}:\ x'_4 = \theta \left(\frac{x'_1}{1-x'_1}\right)^{\phi} \left(x'_2 + \sqrt{x_2'^2+1}\right)^{\alpha} exp\left[\beta x'_3\right]$$

$$S_{UUNL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x_1'^2+1}\right)^{\phi} \left(x'_2 + \sqrt{x_2'^2+1}\right)^{\alpha} exp\left[\beta x'_3\right]$$

$$S_{JJLL}:\ x'_4 = \theta\, exp\left[\phi\, f_1(x'_1)\right] exp\left[\alpha\, f_1(x'_2)\right]\left(x'_3\right)^{\beta}$$

$$S_{NNLL}:\ x'_4 = \theta\, exp\left[\phi x'_1\right] exp\left[\alpha x'_2\right]\left(x'_3\right)^{\beta}$$

$$S_{LNLL}:\ x'_4 = \theta \left(x'_1\right)^{\phi} exp\left[\alpha x'_2\right]\left(x'_3\right)^{\beta}$$

$$S_{BNLL}:\ x'_4 = \theta \left(\frac{x'_1}{1-x'_1}\right)^{\phi} exp\left[\alpha x'_2\right]\left(x'_3\right)^{\beta}$$

$$S_{UNLL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x_1'^2+1}\right)^{\phi} exp\left[\alpha x'_2\right]\left(x'_3\right)^{\beta}$$

$$S_{NLLL}:\ x'_4 = \theta\, exp\left[\phi x'_1\right]\left(x'_2\right)^{\alpha}\left(x'_3\right)^{\beta}$$

$$S_{LLLL}:\ x'_4 = \theta \left(x'_1\right)^{\phi}\left(x'_2\right)^{\alpha}\left(x'_3\right)^{\beta}$$

$$S_{BLLL}:\ x'_4 = \theta \left(\frac{x'_1}{1-x'_1}\right)^{\phi}\left(x'_2\right)^{\alpha}\left(x'_3\right)^{\beta}$$

$$S_{ULLL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x_1'^2+1}\right)^{\phi}\left(x'_2\right)^{\alpha}\left(x'_3\right)^{\beta}$$

$$S_{NBLL}:\ x'_4 = \theta\, exp\left[\phi x'_1\right]\left(\frac{x'_2}{1-x'_2}\right)^{\alpha}\left(x'_3\right)^{\beta}$$

$$S_{LBLL}:\ x'_4 = \theta \left(x'_1\right)^{\phi}\left(\frac{x'_2}{1-x'_2}\right)^{\alpha}\left(x'_3\right)^{\beta}$$

$$S_{BBLL}:\ x'_4 = \theta \left(\frac{x'_1}{1-x'_1}\right)^{\phi}\left(\frac{x'_2}{1-x'_2}\right)^{\alpha}\left(x'_3\right)^{\beta}$$

$$S_{UBLL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x_1'^2+1}\right)^{\phi}\left(\frac{x'_2}{1-x'_2}\right)^{\alpha}\left(x'_3\right)^{\beta}$$

$$S_{NULL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right]\left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{\alpha} \left(x'_3\right)^{\beta}$$

$$S_{LUII}:\ x'_4 = \theta \left(x'_1\right)^{\phi} \left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{\alpha} \left(x'_3\right)^{\beta}$$

$$S_{BULL}:\ x'_4 = \theta \left(\frac{x'_1}{1 - x'_1}\right)^{\phi} \left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{\alpha} \left(x'_3\right)^{\beta}$$

$$S_{UULL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{\phi} \left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{\alpha} \left(x'_3\right)^{\beta}$$

$$S_{JJBL}:\ x'_4 = \theta\, exp\left[\phi\, f_1\left(x'_1\right)\right] exp\left[\alpha\, f_1\left(x'_2\right)\right] \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{NNBL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right] exp\left[\alpha\, x'_2\right] \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{LNBL}:\ x'_4 = \theta \left(x'_1\right)^{\phi}\ exp\left[\alpha\, x'_2\right] \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{BNBL}:\ x'_4 = \theta \left(\frac{x'_1}{1 - x'_1}\right)^{\phi}\ exp\left[\alpha\, x'_2\right] \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{UNBL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{\phi}\ exp\left[\alpha\, x'_2\right] \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{NLBL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right] \left(x'_2\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{LLBL}:\ x'_4 = \theta \left(x'_1\right)^{\phi} \left(x'_2\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{BLBL}:\ x'_4 = \theta \left(\frac{x'_1}{1 - x'_1}\right)^{\phi} \left(x'_2\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{ULBL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{\phi} \left(x'_2\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{NBBL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right] \left(\frac{x'_2}{1 - x'_2}\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{LBBL}:\ x'_4 = \theta \left(x'_1\right)^{\phi} \left(\frac{x'_2}{1 - x'_2}\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{BBBL}:\ x'_4 = \theta \left(\frac{x'_1}{1 - x'_1}\right)^{\phi} \left(\frac{x'_2}{1 - x'_2}\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{UBBL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{\phi} \left(\frac{x'_2}{1 - x'_2}\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{NUBL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right] \left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{LUBL}:\ x'_4 = \theta \left(x'_1\right)^{\phi} \left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{BUBL}:\ x'_4 = \theta \left(\frac{x'_1}{1 - x'_1}\right)^{\phi} \left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{UUBL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{\phi} \left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{\alpha} \left(\frac{x'_3}{1 - x'_3}\right)^{\beta}$$

$$S_{JJUL}:\ x'_4 = \theta\, exp\left[\phi\, f_J(x'_1)\right] exp\left[\alpha\, f_J(x'_2)\right] \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{NNUL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right] exp\left[\alpha\, x'_2\right] \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{LNUL}:\ x'_4 = \theta \left(x'_1\right)^{\phi} exp\left[\alpha\, x'_2\right] \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{BNUL}:\ x'_4 = \theta \left(\frac{x'_1}{1 - x'_1}\right)^{\phi} exp\left[\alpha\, x'_2\right] \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{UNUL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{\phi} exp\left[\alpha\, x'_2\right] \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{NLUL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right] \left(x'_2\right)^{\alpha} \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{LLUL}:\ x'_4 = \theta \left(x'_1\right)^{\phi} \left(x'_2\right)^{\alpha} \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{BLUL}:\ x'_4 = \theta \left(\frac{x'_1}{1 - x'_1}\right)^{\phi} \left(x'_2\right)^{\alpha} \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{ULUL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{\phi} \left(x'_2\right)^{\alpha} \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{NBUL}:\ x'_4 = \theta\, exp\left[\phi\, x'_1\right] \left(\frac{x'_2}{1 - x'_2}\right)^{\alpha} \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{LBUL}:\ x'_4 = \theta \left(x'_1\right)^{\phi} \left(\frac{x'_2}{1 - x'_2}\right)^{\alpha} \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{BBUL}:\ x'_4 = \theta \left(\frac{x'_1}{1 - x'_1}\right)^{\phi} \left(\frac{x'_2}{1 - x'_2}\right)^{\alpha} \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{UBUL}:\ x'_4 = \theta \left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{\phi} \left(\frac{x'_2}{1 - x'_2}\right)^{\alpha} \left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}$$

$$S_{NUUL}:\ x_4' = \theta\, exp\left[\phi\, x_1'\right]\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta}$$

$$S_{LUUL}:\ x_4' = \theta\,(x_1')^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta}$$

$$S_{BUUL}:\ x_4' = \theta\left(\frac{x_1'}{1-x_1'}\right)^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta}$$

$$S_{UUUL}:\ x_4' = \theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta}$$

$$S_{IJNB}:\ x_4' = \left\{1+\theta^{-1}\, exp\left[-\phi\, f_1(x_1')\right] exp\left[-\alpha\, f_2(x_2')\right] exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{NNNB}:\ x_4' = \left\{1+\theta^{-1}\, exp\left[-\phi\, x_1'\right] exp\left[-\alpha\, x_2'\right] exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{LNNB}:\ x_4' = \left\{1+\theta^{-1}\,(x_1')^{-\phi}\, exp\left[-\alpha\, x_2'\right] exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{BNNB}:\ x_4' = \left\{1+\theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi} exp\left[-\alpha\, x_2'\right] exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{UNNB}:\ x_4' = \left\{1+\theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi} exp\left[-\alpha\, x_2'\right] exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{NLNB}:\ x_4' = \left\{1+\theta^{-1}\, exp\left[-\phi\, x_1'\right](x_2')^{-\alpha}\, exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{LLNB}:\ x_4' = \left\{1+\theta^{-1}\,(x_1')^{-\phi}\,(x_2')^{-\alpha}\, exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{BLNB}:\ x_4' = \left\{1+\theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}(x_2')^{-\alpha}\, exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{ULNB}:\ x_4' = \left\{1+\theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}(x_2')^{-\alpha}\, exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{NBNB}:\ x_4' = \left\{1+\theta^{-1}\, exp\left[-\phi\, x_1'\right]\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha} exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{LBNB}:\ x_4' = \left\{1+\theta^{-1}\,(x_1')^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha} exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{BBNB}:\ x_4' = \left\{1+\theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha} exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{UBNB}:\ x_4' = \left\{1+\theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha} exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{NUNB}:\ x_4' = \left\{1+\theta^{-1}\, exp\left[-\phi\, x_1'\right]\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha} exp\left[-\beta\, x_3'\right]\right\}^{-1}$$

$$S_{LUNB}:\ x'_4 = \left\{1+\theta^{-1}\left(x'_1\right)^{-\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha} exp\left[-\beta x'_3\right]\right\}^{-1}$$

$$S_{BUNB}:\ x'_4 = \left\{1+\theta^{-1}\left(\frac{x'_1}{1-x'_1}\right)^{-\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha} exp\left[-\beta x'_3\right]\right\}^{-1}$$

$$S_{UUNB}:\ x'_4 = \left\{1+\theta^{-1}\left(x'_1+\sqrt{x'^2_1+1}\right)^{-\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha} exp\left[-\beta x'_3\right]\right\}^{-1}$$

$$S_{IJLB}:\ x'_4 = \left\{1+\theta^{-1} exp\left[-\phi f_I\left(x'_1\right)\right] exp\left[-\alpha f_J\left(x'_2\right)\right]\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{NNLB}:\ x'_4 = \left\{1+\theta^{-1} exp\left[-\phi x'_1\right] exp\left[-\alpha x'_2\right]\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{LNLB}:\ x'_4 = \left\{1+\theta^{-1}\left(x'_1\right)^{-\phi} exp\left[-\alpha x'_2\right]\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{BNLB}:\ x'_4 = \left\{1+\theta^{-1}\left(\frac{x'_1}{1-x'_1}\right)^{-\phi} exp\left[-\alpha x'_2\right]\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{UNLB}:\ x'_4 = \left\{1+\theta^{-1}\left(x'_1+\sqrt{x'^2_1+1}\right)^{-\phi} exp\left[-\alpha x'_2\right]\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{NLLB}:\ x'_4 = \left\{1+\theta^{-1} exp\left[-\phi x'_1\right]\left(x'_2\right)^{-\alpha}\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{LLLB}:\ x'_4 = \left\{1+\theta^{-1}\left(x'_1\right)^{-\phi}\left(x'_2\right)^{-\alpha}\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{BLLB}:\ x'_4 = \left\{1+\theta^{-1}\left(\frac{x'_1}{1-x'_1}\right)^{-\phi}\left(x'_2\right)^{-\alpha}\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{ULLB}:\ x'_4 = \left\{1+\theta^{-1}\left(x'_1+\sqrt{x'^2_1+1}\right)^{-\phi}\left(x'_2\right)^{-\alpha}\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{NBLB}:\ x'_4 = \left\{1+\theta^{-1} exp\left[-\phi x'_1\right]\left(\frac{x'_2}{1-x'_2}\right)^{-\alpha}\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{LBLB}:\ x'_4 = \left\{1+\theta^{-1}\left(x'_1\right)^{-\phi}\left(\frac{x'_2}{1-x'_2}\right)^{-\alpha}\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{BBLB}:\ x'_4 = \left\{1+\theta^{-1}\left(\frac{x'_1}{1-x'_1}\right)^{-\phi}\left(\frac{x'_2}{1-x'_2}\right)^{-\alpha}\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{UBLB}:\ x'_4 = \left\{1+\theta^{-1}\left(x'_1+\sqrt{x'^2_1+1}\right)^{-\phi}\left(\frac{x'_2}{1-x'_2}\right)^{-\alpha}\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{NULB}:\ x'_4 = \left\{1+\theta^{-1} exp\left[-\phi x'_1\right]\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha}\left(x'_3\right)^{-\beta}\right\}^{-1}$$

$$S_{LULB}:\quad x_4' = \left\{1 + \theta^{-1}\,(x_1')^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}(x_3')^{-\beta}\right\}^{-1}$$

$$S_{BULB}:\quad x_4' = \left\{1 + \theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}(x_3')^{-\beta}\right\}^{-1}$$

$$S_{UULB}:\quad x_4' = \left\{1 + \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}(x_3')^{-\beta}\right\}^{-1}$$

$$S_{IJBB}:\quad x_4' = \left\{1 + \theta^{-1}\,exp\left[-\phi\, f_I(x_1')\right]exp\left[-\alpha\, f_J(x_2')\right]\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}^{-1}$$

$$S_{NNBB}:\quad x_4' = \left\{1 + \theta^{-1}\,exp\left[-\phi\, x_1'\right]exp\left[-\alpha\, x_2'\right]\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}^{-1}$$

$$S_{LNBB}:\quad x_4' = \left\{1 + \theta^{-1}\,(x_1')^{-\phi}\,exp\left[-\alpha\, x_2'\right]\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}^{-1}$$

$$S_{BNBB}:\quad x_4' = \left\{1 + \theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}exp\left[-\alpha\, x_2'\right]\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}^{-1}$$

$$S_{UNBB}:\quad x_4' = \left\{1 + \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}exp\left[-\alpha\, x_2'\right]\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}^{-1}$$

$$S_{NLBB}:\quad x_4' = \left\{1 + \theta^{-1}\,exp\left[-\phi\, x_1'\right](x_2')^{-\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}^{-1}$$

$$S_{LLBB}:\quad x_4' = \left\{1 + \theta^{-1}\,(x_1')^{-\phi}\,(x_2')^{-\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}^{-1}$$

$$S_{BLBB}:\quad x_4' = \left\{1 + \theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}(x_2')^{-\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}^{-1}$$

$$S_{ULBB}:\quad x_4' = \left\{1 + \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}(x_2')^{-\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}^{-1}$$

$$S_{NBBB}:\quad x_4' = \left\{1 + \theta^{-1}\,exp\left[-\phi\, x_1'\right]\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}^{-1}$$

$$S_{LBBB}:\ x'_4 = \left\{1 + \theta^{-1}\left(x'_1\right)^{-\phi}\left(\frac{x'_2}{1 - x'_2}\right)^{-\alpha}\left(\frac{x'_3}{1 - x'_3}\right)^{-\beta}\right\}^{-1}$$

$$S_{BBBB}:\ x'_4 = \left\{1 + \theta^{-1}\left(\frac{x'_1}{1 - x'_1}\right)^{-\phi}\left(\frac{x'_2}{1 - x'_2}\right)^{-\alpha}\left(\frac{x'_3}{1 - x'_3}\right)^{-\beta}\right\}^{-1}$$

$$S_{UBBB}:\ x'_4 = \left\{1 + \theta^{-1}\left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{-\phi}\left(\frac{x'_2}{1 - x'_2}\right)^{-\alpha}\left(\frac{x'_3}{1 - x'_3}\right)^{-\beta}\right\}^{-1}$$

$$S_{NUBB}:\ x'_4 = \left\{1 + \theta^{-1}\ exp\left[-\phi\, x'_1\right]\left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{-\alpha}\left(\frac{x'_3}{1 - x'_3}\right)^{-\beta}\right\}^{-1}$$

$$S_{LUBB}:\ x'_4 = \left\{1 + \theta^{-1}\left(x'_1\right)^{-\phi}\left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{-\alpha}\left(\frac{x'_3}{1 - x'_3}\right)^{-\beta}\right\}^{-1}$$

$$S_{BUBB}:\ x'_4 = \left\{1 + \theta^{-1}\left(\frac{x'_1}{1 - x'_1}\right)^{-\phi}\left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{-\alpha}\left(\frac{x'_3}{1 - x'_3}\right)^{-\beta}\right\}^{-1}$$

$$S_{UUBB}:\ x'_4 = \left\{1 + \theta^{-1}\left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{-\phi}\left(x'_2 + \sqrt{x'^2_2 + 1}\right)^{-\alpha}\left(\frac{x'_3}{1 - x'_3}\right)^{-\beta}\right\}^{-1}$$

$$\mathbf{S_{IJUB}}:\ x'_4 = \left\{1 + \theta^{-1}\ exp\left[-\phi\, f_I\left(x'_1\right)\right] exp\left[-\alpha\, f_J\left(x'_2\right)\right]\left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{-\beta}\right\}^{-1}$$

$$S_{NNUB}:\ x'_4 = \left\{1 + \theta^{-1}\ exp\left[-\phi\, x'_1\right] exp\left[-\alpha\, x'_2\right]\left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{-\beta}\right\}^{-1}$$

$$S_{LNUB}:\ x'_4 = \left\{1 + \theta^{-1}\left(x'_1\right)^{-\phi}\ exp\left[-\alpha\, x'_2\right]\left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{-\beta}\right\}^{-1}$$

$$S_{BNUB}:\ x'_4 = \left\{1 + \theta^{-1}\left(\frac{x'_1}{1 - x'_1}\right)^{-\phi} exp\left[-\alpha\, x'_2\right]\left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{-\beta}\right\}^{-1}$$

$$S_{UNUB}:\ x'_4 = \left\{1 + \theta^{-1}\left(x'_1 + \sqrt{x'^2_1 + 1}\right)^{-\phi} exp\left[-\alpha\, x'_2\right]\left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{\beta}\right\}^{-1}$$

$$S_{NLUB}:\ x'_4 = \left\{1 + \theta^{-1}\ exp\left[-\phi\, x'_1\right]\left(x'_2\right)^{-\alpha}\left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{-\beta}\right\}^{-1}$$

$$S_{LLUB}:\ x'_4 = \left\{1 + \theta^{-1}\left(x'_1\right)^{-\phi}\left(x'_2\right)^{-\alpha}\left(x'_3 + \sqrt{x'^2_3 + 1}\right)^{-\beta}\right\}^{-1}$$

$$S_{BLUB}:\ x_4' = \left\{1+\theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(x_2'\right)^{-\alpha}\left(x_3'+\sqrt{x_3'^2+1}\right)^{-\beta}\right\}^{-1}$$

$$S_{ULUB}:\ x_4' = \left\{1+\theta^{-1}\left(x_1'+\sqrt{x_1'^2+1}\right)^{-\phi}\left(x_2'\right)^{-\alpha}\left(x_3'+\sqrt{x_3'^2+1}\right)^{-\beta}\right\}^{-1}$$

$$S_{NBUB}:\ x_4' = \left\{1+\theta^{-1}\,exp\left[-\phi\,x_1'\right]\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\left(x_3'+\sqrt{x_3'^2+1}\right)^{-\beta}\right\}^{-1}$$

$$S_{LBUB}:\ x_4' = \left\{1+\theta^{-1}\left(x_1'\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\left(x_3'+\sqrt{x_3'^2+1}\right)^{-\beta}\right\}^{-1}$$

$$S_{BBUB}:\ x_4' = \left\{1+\theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\left(x_3'+\sqrt{x_3'^2+1}\right)^{-\beta}\right\}^{-1}$$

$$S_{UBUB}:\ x_4' = \left\{1+\theta^{-1}\left(x_1'+\sqrt{x_1'^2+1}\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\left(x_3'+\sqrt{x_3'^2+1}\right)^{-\beta}\right\}^{-1}$$

$$S_{NUUB}:\ x_4' = \left\{1+\theta^{-1}\,exp\left[-\phi\,x_1'\right]\left(x_2'+\sqrt{x_2'^2+1}\right)^{-\alpha}\left(x_3'+\sqrt{x_3'^2+1}\right)^{-\beta}\right\}^{-1}$$

$$S_{LUUB}:\ x_4' = \left\{1+\theta^{-1}\left(x_1'\right)^{-\phi}\left(x_2'+\sqrt{x_2'^2+1}\right)^{-\alpha}\left(x_3'+\sqrt{x_3'^2+1}\right)^{-\beta}\right\}^{-1}$$

$$S_{BUUB}:\ x_4' = \left\{1+\theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(x_2'+\sqrt{x_2'^2+1}\right)^{-\alpha}\left(x_3'+\sqrt{x_3'^2+1}\right)^{-\beta}\right\}^{-1}$$

$$S_{UUUB}:\ x_4' = \left\{1+\theta^{-1}\left(x_1'+\sqrt{x_1'^2+1}\right)^{-\phi}\left(x_2'+\sqrt{x_2'^2+1}\right)^{-\alpha}\left(x_3'+\sqrt{x_3'^2+1}\right)^{-\beta}\right\}^{-1}$$

$$S_{IJNU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\,exp\left[\phi\,f_i\left(x_1'\right)\right]exp\left[\alpha\,f_j\left(x_2'\right)\right]exp\left[\beta\,x_3'\right]-\\ \theta^{-1}\,exp\left[-\phi\,f_i\left(x_1'\right)\right]exp\left[-\alpha\,f_j\left(x_2'\right)\right]exp\left[-\beta\,x_3'\right]\end{array}\right\}$$

$$S_{NNNU}:\ x_4' = \frac{1}{2}\left\{\theta\,exp\left[\phi\,x_1'\right]exp\left[\alpha\,x_2'\right]exp\left[\beta\,x_3'\right]-\theta^{-1}\,exp\left[-\phi\,x_1'\right]exp\left[-\alpha\,x_2'\right]exp\left[-\beta\,x_3'\right]\right\}$$

$$S_{LNNU}:\ x_4' = \frac{1}{2}\left\{\theta\left(x_1'\right)^{\phi}\,exp\left[\alpha\,x_2'\right]exp\left[\beta\,x_3'\right]-\theta^{-1}\left(x_1'\right)^{-\phi}\,exp\left[-\alpha\,x_2'\right]exp\left[-\beta\,x_3'\right]\right\}$$

$$S_{BNNU}:\ x_4' = \frac{1}{2}\left\{\theta\left(\frac{x_1'}{1-x_1'}\right)^{\phi}exp\left[\alpha\,x_2'\right]exp\left[\beta\,x_3'\right]-\theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}exp\left[-\alpha\,x_2'\right]exp\left[-\beta\,x_3'\right]\right\}$$

$$S_{UNNU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x_1'+\sqrt{x_1'^2+1}\right)^{\phi}exp\left[\alpha\,x_2'\right]exp\left[\beta\,x_3'\right]-\\ \theta^{-1}\left(x_1'+\sqrt{x_1'^2+1}\right)^{-\phi}exp\left[-\alpha\,x_2'\right]exp\left[-\beta\,x_3'\right]\end{array}\right\}$$

$$S_{NLNU}:\ x_4' = \frac{1}{2}\left\{\theta\, exp[\phi x_1'](x_2')^{\alpha}\, exp[\beta x_3'] - \theta^{-1}\, exp[-\phi x_1'](x_2')^{-\alpha}\, exp[-\beta x_3']\right\}$$

$$S_{LLNU}:\ x_4' = \frac{1}{2}\left\{\theta\,(x_1')^{\phi}\,(x_2')^{\alpha}\, exp[\beta x_3'] - \theta^{-1}\,(x_1')^{-\phi}\,(x_2')^{-\alpha}\, exp[-\beta x_3']\right\}$$

$$S_{BLNU}:\ x_4' = \frac{1}{2}\left\{\theta\left(\frac{x_1'}{1-x_1'}\right)^{\phi}(x_2')^{\alpha}\, exp[\beta x_3'] - \theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}(x_2')^{-\alpha}\, exp[-\beta x_3']\right\}$$

$$S_{ULNU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}(x_2')^{\alpha}\, exp[\beta x_3'] - \\ \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}(x_2')^{-\alpha}\, exp[-\beta x_3']\end{array}\right\}$$

$$S_{NBNU}:\ x_4' = \frac{1}{2}\left\{\theta\, exp[\phi x_1']\left(\frac{x_2'}{1-x_2'}\right)^{\alpha} exp[\beta x_3'] - \theta^{-1}\, exp[-\phi x_1']\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha} exp[-\beta x_3']\right\}$$

$$S_{LBNU}:\ x_4' = \frac{1}{2}\left\{\theta\,(x_1')^{\phi}\left(\frac{x_2'}{1-x_2'}\right)^{\alpha} exp[\beta x_3'] - \theta^{-1}\,(x_1')^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha} exp[-\beta x_3']\right\}$$

$$S_{BBNU}:\ x_4' = \frac{1}{2}\left\{\theta\left(\frac{x_1'}{1-x_1'}\right)^{\phi}\left(\frac{x_2'}{1-x_2'}\right)^{\alpha} exp[\beta x_3'] - \theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha} exp[-\beta x_3']\right\}$$

$$S_{UBNU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}\left(\frac{x_2'}{1-x_2'}\right)^{\alpha} exp[\beta x_3'] - \\ \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha} exp[-\beta x_3']\end{array}\right\}$$

$$S_{NUNU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\, exp[\phi x_1']\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha} exp[\beta x_3'] - \\ \theta^{-1}\, exp[-\phi x_1']\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha} exp[-\beta x_3']\end{array}\right\}$$

$$S_{LUNU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\,(x_1')^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha} exp[\beta x_3'] - \\ \theta^{-1}\,(x_1')^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha} exp[-\beta x_3']\end{array}\right\}$$

$$S_{BUNU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(\frac{x_1'}{1-x_1'}\right)^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha} exp[\beta x_3'] - \\ \theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha} exp[-\beta x_3']\end{array}\right\}$$

$$S_{UUNU}\ \ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha} exp[\beta x_3'] - \\ \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha} exp[-\beta x_3']\end{array}\right\}$$

$$S_{JLLU}:\quad x'_4=\frac{1}{2}\left\{\begin{array}{l}\theta\, exp\left[\phi\, f_1(x'_1)\right] exp\left[\alpha\, f_1(x'_2)\right](x'_3)^{\beta}-\\ \theta^{-1}\, exp\left[-\phi\, f_1(x'_1)\right] exp\left[-\alpha\, f_1(x'_2)\right](x'_3)^{-\beta}\end{array}\right\}$$

$$S_{NNLU}:\quad x'_4=\frac{1}{2}\left\{\theta\, exp\left[\phi\, x'_1\right] exp\left[\alpha\, x'_2\right](x'_3)^{\beta}-\theta^{-1}\, exp\left[-\phi\, x'_1\right] exp\left[-\alpha\, x'_2\right](x'_3)^{-\beta}\right\}$$

$$S_{LNLU}:\quad x'_4=\frac{1}{2}\left\{\theta\,(x'_1)^{\phi}\, exp\left[\alpha\, x'_2\right](x'_3)^{\beta}-\theta^{-1}\,(x'_1)^{-\phi}\, exp\left[-\alpha\, x'_2\right](x'_3)^{-\beta}\right\}$$

$$S_{BNLU}:\quad x'_4=\frac{1}{2}\left\{\theta\left(\frac{x'_1}{1-x'_1}\right)^{\phi} exp\left[\alpha\, x'_2\right](x'_3)^{\beta}-\theta^{-1}\left(\frac{x'_1}{1-x'_1}\right)^{-\phi} exp\left[-\alpha\, x'_2\right](x'_3)^{-\beta}\right\}$$

$$S_{UNLU}:\quad x'_4=\frac{1}{2}\left\{\begin{array}{l}\theta\left(x'_1+\sqrt{x'^2_1+1}\right)^{\phi} exp\left[\alpha\, x'_2\right](x'_3)^{\beta}-\\ \theta^{-1}\left(x'_1+\sqrt{x'^2_1+1}\right)^{-\phi} exp\left[-\alpha\, x'_2\right](x'_3)^{-\beta}\end{array}\right\}$$

$$S_{NLLU}:\quad x'_4=\frac{1}{2}\left\{\theta\, exp\left[\phi\, x'_1\right](x'_2)^{\alpha}\,(x'_3)^{\beta}-\theta^{-1}\, exp\left[-\phi\, x'_1\right](x'_2)^{-\alpha}\,(x'_3)^{-\beta}\right\}$$

$$S_{LLLU}:\quad x'_4=\frac{1}{2}\left\{\theta\,(x'_1)^{\phi}\,(x'_2)^{\alpha}\,(x'_3)^{\beta}-\theta^{-1}\,(x'_1)^{-\phi}\,(x'_2)^{-\alpha}\,(x'_3)^{-\beta}\right\}$$

$$S_{BLLU}:\quad x'_4=\frac{1}{2}\left\{\theta\left(\frac{x'_1}{1-x'_1}\right)^{\phi}(x'_2)^{\alpha}\,(x'_3)^{\beta}-\theta^{-1}\left(\frac{x'_1}{1-x'_1}\right)^{-\phi}(x'_2)^{-\alpha}\,(x'_3)^{-\beta}\right\}$$

$$S_{ULLU}:\quad x'_4=\frac{1}{2}\left\{\theta\left(x'_1+\sqrt{x'^2_1+1}\right)^{\phi}(x'_2)^{\alpha}\,(x'_3)^{\beta}-\theta^{-1}\left(x'_1+\sqrt{x'^2+1}\right)^{-\phi}(x'_2)^{-\alpha}\,(x'_3)^{-\beta}\right\}$$

$$S_{NBLU}:\quad x'_4=\frac{1}{2}\left\{\theta\, exp\left[\phi\, x'_1\right]\left(\frac{x'_2}{1-x'_2}\right)^{\alpha}(x'_3)^{\beta}-\theta^{-1}\, exp\left[-\phi\, x'_1\right]\left(\frac{x'_2}{1-x'_2}\right)^{-\alpha}(x'_3)^{-\beta}\right\}$$

$$S_{LBLU}:\quad x'_4=\frac{1}{2}\left\{\theta\,(x'_1)^{\phi}\left(\frac{x'_2}{1-x'_2}\right)^{\alpha}(x'_3)^{\beta}-\theta^{-1}\,(x'_1)^{-\phi}\left(\frac{x'_2}{1-x'_2}\right)^{-\alpha}(x'_3)^{-\beta}\right\}$$

$$S_{BBLU}:\quad x'_4=\frac{1}{2}\left\{\theta\left(\frac{x'_1}{1-x'_1}\right)^{\phi}\left(\frac{x'_2}{1-x'_2}\right)^{\alpha}(x'_3)^{\beta}-\theta^{-1}\left(\frac{x'_1}{1-x'_1}\right)^{-\phi}\left(\frac{x'_2}{1-x'_2}\right)^{-\alpha}(x'_3)^{-\beta}\right\}$$

$$S_{UBLU}:\quad x'_4=\frac{1}{2}\left\{\begin{array}{l}\theta\left(x'_1+\sqrt{x'^2_1+1}\right)^{\phi}\left(\frac{x'_2}{1-x'_2}\right)^{\alpha}(x'_3)^{\beta}-\\ \theta^{-1}\left(x'_1+\sqrt{x'^2_1+1}\right)^{-\phi}\left(\frac{x'_2}{1-x'_2}\right)^{-\alpha}(x'_3)^{-\beta}\end{array}\right\}$$

$$S_{NULU}:\quad x'_4=\frac{1}{2}\left\{\begin{array}{l}\theta\, exp\left[\phi\, x'_1\right]\left(x'_2+\sqrt{x'^2_2+1}\right)^{\alpha}(x'_3)^{\beta}-\\ \theta^{-1}\, exp\left[-\phi\, x'_1\right]\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha}(x'_3)^{-\beta}\end{array}\right\}$$

$$S_{LULU}:\ x'_4 = \frac{1}{2}\left\{\theta\,(x'_1)^{\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{\alpha}(x'_3)^{\beta} - \theta^{-1}\,(x'_1)^{-\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha}(x'_3)^{-\beta}\right\}$$

$$S_{BULU}:\ x'_4 = \frac{1}{2}\left\{\begin{array}{l}\theta\left(\dfrac{x'_1}{1-x'_1}\right)^{\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{\alpha}(x'_3)^{\beta} - \\ \theta^{-1}\left(\dfrac{x'_1}{1-x'_1}\right)^{-\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha}(x'_3)^{-\beta}\end{array}\right\}$$

$$S_{UULU}\ \ x'_4 = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x'_1+\sqrt{x'^2_1+1}\right)^{\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{\alpha}(x'_3)^{\beta} - \\ \theta^{-1}\left(x'_1+\sqrt{x'^2_1+1}\right)^{-\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha}(x'_3)^{-\beta}\end{array}\right\}$$

$$S_{IJBU}:\ x'_4 = \frac{1}{2}\left\{\begin{array}{l}\theta\, exp[\phi\, f_1(x'_1)]\, exp[\alpha\, f_J(x'_2)]\left(\dfrac{x'_3}{1-x'_3}\right)^{\beta} - \\ \theta^{-1}\, exp[-\phi\, f_1(x'_1)]\, exp[-\alpha\, f_J(x'_2)]\left(\dfrac{x'_3}{1-x'_3}\right)^{-\beta}\end{array}\right\}$$

$$S_{NNBU}:\ x'_4 = \frac{1}{2}\left\{\theta\, exp[\phi\, x'_1]\, exp[\alpha\, x'_2]\left(\frac{x'_3}{1-x'_3}\right)^{\beta} - \theta^{-1}\, exp[-\phi\, x'_1]\, exp[-\alpha\, x'_2]\left(\frac{x'_3}{1-x'_3}\right)^{-\beta}\right\}$$

$$S_{LNBU}:\ x'_4 = \frac{1}{2}\left\{\theta\,(x'_1)^{\phi}\, exp[\alpha\, x'_2]\left(\frac{x'_3}{1-x'_3}\right)^{\beta} - \theta^{-1}\,(x'_1)^{-\phi}\, exp[-\alpha\, x'_2]\left(\frac{x'_3}{1-x'_3}\right)^{-\beta}\right\}$$

$$S_{BNBU}:\ x'_4 = \frac{1}{2}\left\{\theta\left(\frac{x'_1}{1-x'_1}\right)^{\phi} exp[\alpha\, x'_2]\left(\frac{x'_3}{1-x'_3}\right)^{\beta} - \theta^{-1}\left(\frac{x'_1}{1-x'_1}\right)^{-\phi} exp[-\alpha\, x'_2]\left(\frac{x'_3}{1-x'_3}\right)^{-\beta}\right\}$$

$$S_{UNBU}:\ x'_4 = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x'_1+\sqrt{x'^2_1+1}\right)^{\phi} exp[\alpha\, x'_2]\left(\dfrac{x'_3}{1-x'_3}\right)^{\beta} - \\ \theta^{-1}\left(x'_1+\sqrt{x'^2_1+1}\right)^{-\phi} exp[-\alpha\, x'_2]\left(\dfrac{x'_3}{1-x'_3}\right)^{-\beta}\end{array}\right\}$$

$$S_{NLBU}:\ x'_4 = \frac{1}{2}\left\{\theta\, exp[\phi\, x'_1](x'_2)^{\alpha}\left(\frac{x'_3}{1-x'_3}\right)^{\beta} - \theta^{-1}\, exp[-\phi\, x'_1](x'_2)^{-\alpha}\left(\frac{x'_3}{1-x'_3}\right)^{\beta}\right\}$$

$$S_{LLBU}:\ x'_4 = \frac{1}{2}\left\{\theta\,(x'_1)^{\phi}\,(x'_2)^{\alpha}\left(\frac{x'_3}{1-x'_3}\right)^{\beta} - \theta^{-1}\,(x'_1)^{-\phi}\,(x'_2)^{-\alpha}\left(\frac{x'_3}{1-x'_3}\right)^{-\beta}\right\}$$

$$S_{BLBU}:\ x'_4 = \frac{1}{2}\left\{\theta\left(\frac{x'_1}{1-x'_1}\right)^{\phi}(x'_2)^{\alpha}\left(\frac{x'_3}{1-x'_3}\right)^{\beta} - \theta^{-1}\left(\frac{x'_1}{1-x'_1}\right)^{-\phi}(x'_2)^{-\alpha}\left(\frac{x'_3}{1-x'_3}\right)^{-\beta}\right\}$$

$$S_{ULBU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}(x_2')^{\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{\beta} - \\ \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}(x_2')^{-\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{-\beta}\end{array}\right\}$$

$$S_{NBBU}:\ x_4' = \frac{1}{2}\left\{\theta\, exp\left[\phi\, x_1'\right]\left(\frac{x_2'}{1-x_2'}\right)^{\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{\beta} - \theta^{-1}\, exp\left[-\phi\, x_1'\right]\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}$$

$$S_{LBBU}:\ x_4' = \frac{1}{2}\left\{\theta\,(x_1')^{\phi}\left(\frac{x_2'}{1-x_2'}\right)^{\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{\beta} - \theta^{-1}(x_1')^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}$$

$$S_{BBBU}:\ x_4' = \frac{1}{2}\left\{\theta\left(\frac{x_1'}{1-x_1'}\right)^{\phi}\left(\frac{x_2'}{1-x_2'}\right)^{\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{\beta} - \theta^{-1}\left(\frac{x_1'}{1-x_1'}\right)^{-\phi}\left(\frac{x_2'}{1-x_2'}\right)^{-\alpha}\left(\frac{x_3'}{1-x_3'}\right)^{-\beta}\right\}$$

$$S_{UBBU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}\left(\dfrac{x_2'}{1-x_2'}\right)^{\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{\beta} - \\ \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}\left(\dfrac{x_2'}{1-x_2'}\right)^{-\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{-\beta}\end{array}\right\}$$

$$S_{NUBU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\, exp\left[\phi\, x_1'\right]\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{\beta} - \\ \theta^{-1}\, exp\left[-\phi\, x_1'\right]\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{-\beta}\end{array}\right\}$$

$$S_{LUBU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\,(x_1')^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{\beta} - \\ \theta^{-1}(x_1')^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{-\beta}\end{array}\right\}$$

$$S_{BUBU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(\dfrac{x_1'}{1-x_1'}\right)^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{\beta} - \\ \theta^{-1}\left(\dfrac{x_1'}{1-x_1'}\right)^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{-\beta}\end{array}\right\}$$

$$S_{UUBU}:\ x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{\beta} - \\ \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}\left(x_2' + \sqrt{x_2'^2+1}\right)^{-\alpha}\left(\dfrac{x_3'}{1-x_3'}\right)^{-\beta}\end{array}\right\}$$

$$S_{JJUU}:\quad x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\, exp\left[\phi\, f_1(x_1')\right] exp\left[\alpha\, f_1(x_2')\right]\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta} - \\ \theta^{-1}\, exp\left[-\phi\, f_1(x_1')\right] exp\left[-\alpha\, f_1(x_2')\right]\left(x_3' + \sqrt{x_3'^2+1}\right)^{-\beta}\end{array}\right\}$$

$$S_{NNUU}:\quad x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\, exp\left[\phi\, x_1'\right] exp\left[\alpha\, x_2'\right]\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta} - \\ \theta^{-1}\, exp\left[-\phi\, x_1'\right] exp\left[-\alpha\, x_2'\right]\left(x_3' + \sqrt{x_3'^2+1}\right)^{-\beta}\end{array}\right\}$$

$$S_{LNUU}:\quad x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\, (x_1')^{\phi}\, exp\left[\alpha\, x_2'\right]\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta} - \\ \theta^{-1}\, (x_1')^{-\phi}\, exp\left[-\alpha\, x_2'\right]\left(x_3' + \sqrt{x_3'^2+1}\right)^{-\beta}\end{array}\right\}$$

$$S_{BNUU}:\quad x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(\dfrac{x_1'}{1-x_1'}\right)^{\phi} exp\left[\alpha\, x_2'\right]\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta} - \\ \theta^{-1}\left(\dfrac{x_1'}{1-x_1'}\right)^{-\phi} exp\left[-\alpha\, x_2'\right]\left(x_3' + \sqrt{x_3'^2+1}\right)^{-\beta}\end{array}\right\}$$

$$S_{UNUU}:\quad x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi} exp\left[\alpha\, x_2'\right]\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta} - \\ \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi} exp\left[-\alpha\, x_2'\right]\left(x_3' + \sqrt{x_3'^2+1}\right)^{-\beta}\end{array}\right\}$$

$$S_{NLUU}:\quad x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\, exp\left[\phi\, x_1'\right](x_2')^{\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta} - \\ \theta^{-1}\, exp\left[-\phi\, x_1'\right](x_2')^{-\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{-\beta}\end{array}\right\}$$

$$S_{LLUU}:\quad x_4' = \frac{1}{2}\left\{\theta\, (x_1')^{\phi}\, (x_2')^{\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta} - \theta^{-1}\, (x_1')^{-\phi}\, (x_2')^{-\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{-\beta}\right\}$$

$$S_{BLUU}:\quad x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(\dfrac{x_1'}{1-x_1'}\right)^{\phi}(x_2')^{\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta} - \\ \theta^{-1}\left(\dfrac{x_1'}{1-x_1'}\right)^{-\phi}(x_2')^{-\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{-\beta}\end{array}\right\}$$

$$S_{ULUU}:\quad x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\left(x_1' + \sqrt{x_1'^2+1}\right)^{\phi}(x_2')^{\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta} - \\ \theta^{-1}\left(x_1' + \sqrt{x_1'^2+1}\right)^{-\phi}(x_2')^{-\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{-\beta}\end{array}\right\}$$

$$S_{NBUU}:\quad x_4' = \frac{1}{2}\left\{\begin{array}{l}\theta\, exp\left[\phi\, x_1'\right]\left(\dfrac{x_2'}{1-x_2'}\right)^{\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{\beta} - \\ \theta^{-1}\, exp\left[-\phi\, x_1'\right]\left(\dfrac{x_2'}{1-x_2'}\right)^{-\alpha}\left(x_3' + \sqrt{x_3'^2+1}\right)^{-\beta}\end{array}\right\}$$

$$S_{LBUU}:\ x'_4 = \frac{1}{2}\left\{\begin{array}{l} \theta\,(x'_1)^{\phi}\left(\dfrac{x'_2}{1-x'_2}\right)^{\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{\beta} - \\ \theta^{-1}\,(x'_1)^{-\phi}\left(\dfrac{x'_2}{1-x'_2}\right)^{-\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{-\beta} \end{array}\right\}$$

$$S_{BBUU}:\ x'_4 = \frac{1}{2}\left\{\begin{array}{l} \theta\left(\dfrac{x'_1}{1-x'_1}\right)^{\phi}\left(\dfrac{x'_2}{1-x'_2}\right)^{\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{\beta} - \\ \theta^{-1}\left(\dfrac{x'_1}{1-x'_1}\right)^{-\phi}\left(\dfrac{x'_2}{1-x'_2}\right)^{-\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{-\beta} \end{array}\right\}$$

$$S_{UBUU}:\ x'_4 = \frac{1}{2}\left\{\begin{array}{l} \theta\left(x'_1+\sqrt{x'^2_1+1}\right)^{\phi}\left(\dfrac{x'_2}{1-x'_2}\right)^{\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{\beta} - \\ \theta^{-1}\left(x'_1+\sqrt{x'^2_1+1}\right)^{-\phi}\left(\dfrac{x'_2}{1-x'_2}\right)^{-\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{-\beta} \end{array}\right\}$$

$$S_{NUUU}:\ x'_4 = \frac{1}{2}\left\{\begin{array}{l} \theta\, exp\left[\phi\, x'_1\right]\left(x'_2+\sqrt{x'^2_2+1}\right)^{\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{\beta} - \\ \theta^{-1}\, exp\left[-\phi\, x'_1\right]\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{-\beta} \end{array}\right\}$$

$$S_{LUUU}:\ x'_4 = \frac{1}{2}\left\{\begin{array}{l} \theta\,(x'_1)^{\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{\beta} - \\ \theta^{-1}\,(x'_1)^{-\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{-\beta} \end{array}\right\}$$

$$S_{BUUU}:\ x'_4 = \frac{1}{2}\left\{\begin{array}{l} \theta\left(\dfrac{x'_1}{1-x'_1}\right)^{\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{\beta} - \\ \theta^{-1}\left(\dfrac{x'_1}{1-x'_1}\right)^{-\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{-\beta} \end{array}\right\}$$

$$S_{UUUU}:\ x'_4 = \frac{1}{2}\left\{\begin{array}{l} \theta\left(x'_1+\sqrt{x'^2_1+1}\right)^{\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{\beta} - \\ \theta^{-1}\left(x'_1+\sqrt{x'^2_1+1}\right)^{-\phi}\left(x'_2+\sqrt{x'^2_2+1}\right)^{-\alpha}\left(x'_3+\sqrt{x'^2_3+1}\right)^{-\beta} \end{array}\right\}$$

APPENDIX F

Copyright Registration Application Form

STATE UN

5 01

SHORT FORM TX

For a Nondramatic Literary Work
UNITED STATES COPYRIGHT OFFICE

Registration Number

TX TXU

Effective Date of Registration

Application Received

Examined By

Deposit Received
One Two

Correspondence

Fee Received

TYPE OR PRINT IN BLACK INK. DO NOT WRITE ABOVE THIS LINE.

Title of This Work: Alternative title or title of larger work in which this work was published	1	MULTIVARIATE NON-NORMAL PROCESS CAPABILITY INDICES: A SIMULATION APPROACH
Name and Address of Author and Owner of the Copyright: Nationality or domicile Phone, fax, and email	2	Allen Lynn Lewis 806 Rock Hollow Ct. Stillwater, OK 74074 Phone (405) 743-7542 Fax (405) 743-5379 Email Allen lewis@mercmarine.com
Year of Creation:	3	1998
If work has been published, Date and Nation of Publication:	4	a. Date ___ Month ___ Day ___ Year ___ (Month, day, and year all required) b. Nation
Type of Authorship in This Work: Check all that this author created	5	☒ Text (includes fiction, nonfiction, poetry, computer programs, etc.) ☐ Illustrations ☐ Photographs ☐ Compilation of terms or data
Signature: Registration cannot be completed without a signature	6	*I certify that the statements made by me in this application are correct to the best of my knowledge.* Check one ☒ Author ☐ Authorized agent X Allen L. Lewis
OPTIONAL Name and Address of Person to Contact for Rights and Permissions: Phone, fax, and email	7	☒ Check here if same as #2 above Phone ( ) Fax ( ) Email

8 Certificate will be mailed in window envelope to this address:	Name ▼ Allen Lynn Lewis Number/Street/Apt ▼ 806 Rock Hollow Ct. City/State/ZIP ▼ Stillwater, Oklahoma 74074	Complete this space only if you currently hold a Deposit Account in the Copyright Office. 9 Deposit Account # Name DO NOT WRITE HERE Page 1 of ___ pages

17 U.S.C. § 506(e): Any person who knowingly makes a false representation of a material fact in the application for copyright registration provided for by section 409, or in any written statement filed in connection with the application, shall be fined not more than $2,500.
September 1997—100,000

U.S. GOVERNMENT PRINTING OFFICE: 1997-413-750

VITA

Allen L. Lewis

Candidate for the Degree of

Master of Science

Thesis: MULTIVARIATE NON-NORMAL PROCESS CAPABILITY INDICES: A SIMULATION APPROACH

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in New Hampton, Iowa, on June 2, 1968, the son of Gerald and Lorraine Lewis. Married Carolyn Billhorn on August 1, 1992. First son, Nicholas, born on April 6, 1995 and second son, Carston, born on July 24, 1998.

Education: Graduated from LaPorte City High School, LaPorte City, Iowa in May 1986; received Bachelor of Science degree in AeroSpace Engineering from Iowa State University, Ames, Iowa in May 1992. Began graduate study at the University of Nebraska Lincoln, Lincoln, Nebraska in 1995. Completed the requirements for the Master of Science degree at Oklahoma State University, Stillwater, Oklahoma in December, 1998.

Experience: Employed as a welder, machinist, and fabricator during the summers and breaks of my high school and undergraduate years at a small Iowa business owned and operated by my uncle and grandfather. Employed by Iowa State University as both a resident assistant in the residence halls and a recreational area attendant at the Iowa State University Memorial Union. After five years of Air Force R.O.T.C. at Iowa State University, I was commissioned a second lieutenant in the United States Air Force in 1992. My active duty was spent stationed at Vandenberg AFB, CA, where I worked in the missile launch career field. Was employed by Appleton Electric Company in Columbus, Nebraska as a Quality Engineer in 1994. Employed by MerCruiser in Stillwater, Oklahoma as a Quality Engineer, 1996 to present.

Professional Memberships, Certifications, and Honor Societies: American Society for Quality, Certified Reliability Engineer, Alpha Pi Mu.