9-1996

Computational Algebra Applications in Reliability

G Hartless

Lawrence Leemis
William & Mary, lmleem@wm.edu

Recommended Citation

Hartless, G and Leemis, Lawrence, Computational Algebra Applications in Reliability (1996). *IEEE Transactions on Reliability*, 45(3), 393-399.
https://doi.org/10.1109/24.536991

Computational Algebra Applications in Reliability Theory

Glen Hartless
University of Florida
Gainesville

Lawrence Leemis
The College of William & Mary
Williamsburg

Key Words — Computational algebra language, Mathematica, Model selection, Reliability bounds, Symbolic algebra language, Weibull distribution.

Summary & Conclusions — Reliability analysts are typically forced to choose between using an 'algorithmic programming language' or a 'reliability package' for analyzing their models and lifetime data. This paper shows that computational languages can be used to bridge the gap to combine the flexibility of a programming language with the ease of use of a package. Computational languages facilitate the development of new statistical techniques and are excellent teaching tools. This paper considers three diverse reliability problems that are handled easily with a computational algebra language: system reliability bounds, lifetime data analysis, and model selection.

1. INTRODUCTION

Computational algebra languages are tools for solving a wide array of mathematical problems. These interactive frameworks have powerful symbolic & graphical capabilities that are easily & quickly implemented, making these languages superior to standard algorithmic languages such as C or FORTRAN for certain applications. In addition, they provide low-level programming constructs that allow flexibility that is often not available in specific application software packages.

This paper documents our application of one such program, the Mathematica system [4], as a tool. In short, how would one fare in applying such a language to some of the diverse computational problems in reliability? What features of the system make it superior (or inferior) to other programming environments? Can a reliability engineer with a modest computer programming background make use of Mathematica to solve problems efficiently? In order to answer these questions, we chose three problems that test the usefulness of this tool: reliability bounds, lifetime data analysis, and model selection.

Acronyms

MLE	maximum likelihood estimator
KM/PL	Kaplan–Meier product limit.

Notation

$r(\mathbf{p})$	system reliability
$\mathbf{p} = (p_1, p_2, ..., p_n)'$	vector of component reliabilities
P_1, P_2, ..., P_s	minimal path sets
C_1, C_2, ..., C_k	minimal cut sets
$S(t)$	survivor function
λ	Weibull scale parameter
κ	Weibull shape parameter
t_1, t_2, ..., t_n	failure times

$c_1, c_2, \ldots, c_n$	censoring times
$x_i = \min\{t_i, c_i\}$	time on test for item i, $i = 1, 2, \ldots, n$
U	index set of uncensored observations
r	number of observed failures
$L(.)$	likelihood function
$\hat{\lambda}$	Weibull scale parameter MLE
$\hat{\kappa}$	Weibull shape parameter MLE
$h(t)$	hazard function
$H(t)$	cumulative hazard function
$R(t)$	risk set at time t
$n(t)$	cardinality of $R(t)$
$\gamma = \frac{\sigma}{\mu}$	coefficient of variation
$\gamma_3 = E\left[\left(\frac{T-\mu}{\sigma}\right)^3\right]$	skewness
$\gamma_4 = E\left[\left(\frac{T-\mu}{\sigma}\right)^4\right]$	kurtosis.

2. BACKGROUND

Mathematica is an environment for numerical & symbolic computations, with excellent graphics capabilities. It combines the 'interactive user interface' and 'functional programming' of a high level language such as S-Plus with commands & functions that support symbolic manipulation. We will not emphasize the actual coding of our functions, but rather the insight gained from applying Mathematica to reliability problems. The Mathematica code is available from the authors.

Each subsequent section is divided into four parts: an introduction to the particular topic, the implementation of Mathematica to solve the problem, a specific example to illustrate the implementation, and a brief discussion of the merits of the Mathematica solution.

3. RELIABILITY BOUNDS

The reliability $r(\mathbf{p})$ of a s-coherent system of s-independent components can be bounded using minimal path sets and minimal cut sets (Barlow and Proschan [1], p. 37), where $\mathbf{p} = (p_1, p_2, ..., p_n)'$ is a vector of component reliabilities. For a s-coherent system of s-independent components with minimal path sets P_1, P_2, ..., P_s, and minimal cut sets C_1, C_2, ..., C_k, one such bound is

$$\prod_{j=1}^{k}\left[1-\prod_{i\in C_j}(1 - p_i)\right] \le r(\mathbf{p}) \le 1 - \prod_{j=1}^{s}\left[1 - \prod_{i\in P_j} p_i\right],$$

or

$$\prod_{j=1}^{k}\coprod_{i\in C_j} p_i \le r(\mathbf{p}) \le \coprod_{j=1}^{s}\prod_{i\in P_j} p_i.$$

The bounded expression is the probability that the system is operational at one particular time, i.e., the system reliability.

3.1 Mathematica Implementation

Consider a scenario where each component has the same reliability p, where $0 < p < 1$. For various values of p, how precise are our bounds for an arbitrary system configuration? A Mathematica function `bound` calculates and plots the reliability bounds. Since the code for this function is short, it is included below.

```
bound[paths_List, cuts_List] :=
(
  numpaths = Length[paths];
  numcuts = Length[cuts];
  upperbnd = lowerbnd = 1;
  Do[upperbnd = upperbnd * (1 - p ^ Length[paths[[i]]]),{i, numpaths}];
  upperbnd = 1 - upperbnd;
  Do[lowerbnd = lowerbnd * (1 - (1 - p) ^ Length[cuts[[i]]]), {i, numcuts}];
```

```
  Plot[{lowerbnd, upperbnd}, {p, 0, 1}, PlotLabel ->
    "Reliability Bounds", AxesLabel -> {"p", "r(p)"}];
)
```

The arguments passed to `bound` are the *set* of minimal cut sets, and the *set* of minimal path sets.[1]

3.2 Example

To illustrate the use of these reliability bounds on a small system, consider the four-component system in Figure 1. This system has three minimal path sets

$$\{1\}, \{2, 3\}, \{2, 4\}$$

and two minimal cut sets

$$\{1, 2\}, \{1, 3, 4\}.$$

Once `bound` been input, the appropriate Mathematica commands to determine the reliability bounds for this system are

```
In[1]:= mpaths = { {1}, {2, 3}, {2, 4} }
In[2]:= mcuts = { {1, 2}, {1, 3, 4} }
In[3]:= bound[mpaths, mcuts]
```

The upper and lower bounds for the system reliability for components with identical reliabilities are plotted in Figure 2. We can request the expression for, say, the lowerbound, and evaluate it for $p = 0.7$.

[1] One Mathematica data structure is a *list*, which is a set of numbers, algebraic expressions, functions, graphical elements, etc.

```
In[4]:= lowerbnd

                       2              3
Out[4]= (1 - (1 - p) )(1 - (1 - p) )


In[5]:= lowerbnd /. p -> 0.7


Out[5]= 0.88543
```

3.3 Discussion

Mathematica has more power & flexibility than is really needed to solve this problem. Still, the ability to view & experiment with the upper and lower bounds is an educational tool since students can easily explore the behavior of the inequality for different reliabilities and system configurations.

4. LIFETIME DATA ANALYSIS

Consider fitting statistical models to a set of lifetime data. In particular, we wish to fit the Weibull distribution to a right-censored data set using maximum likelihood estimation (the Weibull distribution was chosen because its MLEs must be calculated by numerical methods). Also, we would like to obtain some measures of model adequacy: s-confidence regions for the parameters, a visual comparison of the corresponding non-parametric model, and the ability to compare the fit to that of other parametric models.

4.1 Mathematica Implementation

Recall that the survivor function of the Weibull distribution is

$$S(t) = e^{-(\lambda t)^{\kappa}}$$

for all $t > 0$, $\lambda > 0$, and $\kappa > 0$. Most of the derivation shown below is given in Cox and Oakes [2].

For failure times $t_1, t_2, ..., t_n$ and censoring times $c_1, c_2, ..., c_n$, the log likelihood function can be expressed in terms of the hazard function $h(t)$ and the cumulative hazard function $H(t)$

$$\log \ L(\lambda, \kappa) \ = \ \sum_{i \in U} \log h(x_i, \lambda, \kappa) - \sum_{i=1}^{n} H(x_i, \lambda, \kappa)$$

where U is the set of indices corresponding to uncensored observations, r is the number of observed failures, and $x_i = \min\{t_i, c_i\}$, for $i = 1, 2, \ldots, n$.

Maximum likelihood estimates can be obtained in the usual way by taking the first partial derivatives of the log likelihood function with respect to λ and κ, setting the resulting equations equal to zero, and solving for the parameters.

Given the survivor function, Mathematica can perform all of the *derivations* necessary to obtain this 2×2 system of equations. Given suitable initial estimates, it can solve the equations via Newton's Method. We found that simply guessing the initial estimates resulted in a rather slow convergence of the root solver. Therefore, we used a least squares approach to obtain initial estimates for λ and κ. For the Weibull distribution

$$\log[-\log S(t)] = \kappa \log \lambda + \kappa \log t.$$

Let $R(t)$ be the set of indices corresponding to items at risk at or just prior to time t and let $n(t) = |R(t)|$ be the number of elements in, or the cardinality of $R(t)$. Estimating the survivor function with $\frac{n(t)}{n+1}$, we perform a simple linear regression on $\log x_i$ vs. $\log \left[-\log \frac{n(t)}{n+1}\right]$. The slope of the fitted model is an estimate of κ and the intercept is an estimate of $\kappa \log \lambda$. This is equivalent to a plot of the data on Weibull paper, and has the side benefit of being able to assess model adequacy. By plotting the data and the fitted regression line, we can determine whether the relationship is linear, thereby confirming the appropriateness of the model over the region of the data. There is no difficulty in implementing either the regression or the plot in Mathematica[2].

Initial estimates in hand, Mathematica calculates the maximum of the log likelihood function and plots the log likelihood function surface. The observed information matrix, the asymptotic variance-covariance matrix of the MLEs, and the value of the log likelihood's maximum are all printed. The latter can be used to compare the fits of different families of distributions.

As a further test of the Weibull fit,

$$2[\log L(\hat{\lambda}, \hat{\kappa}) - \log L(\lambda, \kappa)]$$

is asymptotically χ^2_2, and hence we can obtain an approximate s-confidence region for the parameter estimates. This s-confidence region can be used to determine whether the additional parameter in the Weibull distribution (compared to the exponential distribution without a shape parameter) is warranted. If the line $\kappa = 1$ is interior to the s-confidence region, the extra parameter is not s-significant, and the reduced model might be appropriate. Quantiles of the chi-square distribution, and those of many other distributions, are available in Mathematica. Using the Mathematica function `ContourPlot` produces a contour plot of the log likelihood function, which is the boundary of an asymptotically valid s-confidence region for the parameters. The output given in the example that follows represents a balance between precision and computing time. Finally, a graph of the estimated survivor function is produced, along with the Kaplan– Meier product-limit estimator of the survivor function.

[2]Obtaining the Weibull MLEs can be reduced to a one-dimensional search. In order to facilitate the fitting of other two-parameter distributions with the same code, however, this approach was not taken. Additionally, a two-dimensional search was a better test of Mathematica's abilities.

4.2 Example

The data below are ball bearing failure times (in 10^6 revolutions) from Lawless [3]

17.88 28.92 33.00 41.52 42.12 45.60 48.48 51.84 51.96 54.12 55.56 67.80
68.64 68.64 68.88 84.12 93.12 98.64 105.12 105.84 127.92 128.04 173.40.

The commands necessary to run our function `ReliabilityFit` are

```
In[6]: data={{17.88,  1}, {28.92,  1}, {33.00,  1}, {41.52,  1},
{42.12,  1}, {45.60,  1}, {48.48,  1}, {51.84,  1}, {51.96,  1},
{54.12,  1}, {55.56,  1}, {67.80,  1}, {68.64,  1}, {68.64,  1},
{68.88,  1}, {84.12,  1}, {93.12,  1}, {98.64,  1}, {105.12, 1},
{105.84, 1}, {127.92, 1}, {128.04, 1}, {173.40, 1}}
In[7]: SurvFunctWeibull = N[E] ^ (- (LAM x) ^ K)
In[8]: ReliabilityFit[data, WeibullSurvivor, InitEstimWeibull[data]]
```

The second element of each data pair gives its censoring status (0: right censored, 1: observed), and `InitEstimWeibull` is a function that returns initial estimates of the Weibull parameters. The output of the program includes a scatterplot and fitted regression line (Figure 3), a 3-dimensional graph of the log likelihood function (Figure 4), 90% and 95% confidence regions for the MLEs (Figure 5), and a comparison of the MLE survivor function to the corresponding Kaplan–Meier product-limit estimate (Figure 6). The function prints the survivor function, maximum likelihood estimators, the maximum value of the log likelihood function, the observed information matrix, and the asymptotic variance-covariance matrix of the MLEs, as shown here.

```
                             k
                      -(lam t)
Survivor Function: E

Maximum Likelihood Estimators-
```

```
      kappa: 2.102 lambda: 0.01221

Log Likelihood at the MLE: -113.7

Observed Information Matrix:
681327.   874.6

874.6     10.38

Inverse Information Matrix (Variance-Covariance Matrix of the MLEs):
       -6
1.646 10      -0.0001387

-0.0001387   0.1080
```

Two comments with respect to the s-confidence region in Figure 5 follow. First, Figure 5 verifies that a Weibull distribution is a better fit than an exponential distribution since the line $\kappa = 1$ is not interior to the s-confidence region. Second, adjusting an argument in `ReliabilityFit` gives smoother contours at the cost of larger computation times.

4.3 Discussion

Mathematica's overall performance on this larger, more complex problem was adequate. We were pleased with the program's ability to overlay several graphic images such that the scaling & appearance were appropriate. We accomplished nearly everything we had hoped to, although much trial & error was often necessary.

Another advantage of our Mathematica implementation is that our function has only three arguments: the data set, a survivor function, and initial parameter estimates. The log likelihood function, score vector, and information matrix are all determined by Mathematica. As such, the code appears much more like the mathematical derivations than the corresponding code in an algorithmic language. Other two-parameter distributions could be fitted in a similar fashion.

This larger programming application revealed a down-side of using Mathematica. The program syntax can be quite tedious. Slight deviations from the desired command can produce a wealth of syntax errors and warnings, and/or incorrect results.

5. MODEL SELECTION

We select a parametric model to describe a set of lifetime data. In order to limit the number of models, sample moments can be compared to the theoretical moments of the distribution in question.

Two graphical methods used for this purpose are plots of skewness (γ_3) vs. coefficient of variation (γ), and kurtosis (γ_4) vs. skewness (see Cox and Oakes [2], p. 27). The first plot is a visual representation of symmetry and spread, while the second is of peakedness and symmetry. Some distributions, such as the exponential and uniform distributions, reduce to a single point on these plots. Close proximity of the curve to the corresponding sample moments indicates that the distribution is not a bad potential parametric model.

5.1 Mathematica Implementation

Since Mathematica is capable of producing a 3-dimensional parametric plot, it occurred to us that a plot of $(\gamma, \gamma_3, \gamma_4)$ would be of similar use. Unfortunately, the simplicity of the curves makes their proximity to the sample moments difficult to discern without rotating the 3-dimensional plot. One solution to this problem is to find the minimal Euclidean distance between $(\hat{\gamma},\ \hat{\gamma}_3,\ \hat{\gamma}_4)$ and the 3-dimensional moment curves, where $\hat{\gamma}$ is the ratio of the sample standard deviation to the sample mean and

$$\hat{\gamma} = \frac{s}{\bar{t}} \qquad \hat{\gamma}_3 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{t_i - \bar{t}}{s}\right)^3 \qquad \hat{\gamma}_4 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{t_i - \bar{t}}{s}\right)^4$$

The moment curves given here are the function of, at most, a single shape parameter. The value of this parameter when the Euclidean distance is minimized can be used as an initial estimate for obtaining its maximum likelihood estimate. Hence, not only can we see which distributions are possible models, we can also learn something about their parameter estimates. Again, due to Mathematica's root-solving ability, this minimizer was not difficult to develop. Unlike the likelihood function in section 4, the distance functions do not need a particularly accurate initial estimate to converge.

5.2 Example

Consider again the ball bearing data from section 4.2. Its moments are $\hat{\gamma} = 0.52, \hat{\gamma}_3 = 0.88, \hat{\gamma}_4 = 3.19$. Figures 7 & 8 are the skewness vs. coefficient of variation and kurtosis vs. skewness plots for several lifetime distributions: gamma, log logistic, log normal, Pareto, and Weibull. The exponential distribution is a special case of the gamma and Weibull, and has moments $\gamma = 1, \gamma_3 = 2, \gamma_4 = 9$. Also, the uniform distribution has moments $\gamma = \frac{1}{\sqrt{3}}, \gamma_3 = 0, \gamma_4 = \frac{9}{5}$. The sample moments of the ball bearing data are indicated by a point on each graph. Figures 9 & 10 show the 3-dimensional moment curves from two perspectives. Figure 9 looks down at the kurtosis vs. skewness plot, with depth provided by the coefficient of variation. Figure 10 is a rotated version with the kurtosis (height) scale reduced. As explained previously, the behavior is difficult to see.

What follows is the results of determining the minimum Euclidean distance between $(\hat{\gamma},\ \hat{\gamma_3},\ \hat{\gamma_4})$ and each of the moment curves in three dimensions:

```
In[9]: MomentDistance[data]


Out[9]//MatrixForm=


>   DISTRIBUTION         MIN DISTANCE         INITIAL ESTIMATE
```

Uniform	1.64658	NoShapeParameter
Exponential	5.93626	NoShapeParameter
Weibull	0.247013	k -> 1.95794
Gamma	0.488128	k -> 13.0483
LogLogistic	1.38079	k -> 55.9427
Pareto	5.91096	b -> 1737.07
LogNormal	2.9733	sigma -> 0.0314362

5.3 Discussion

It should be kept in mind that the Euclidean distance technique should only be used to determine which models would be obviously inferior. It is not a method with which to choose a model. For instance, in testing this method using randomly generated samples from a log normal distribution, the fitted Weibull and gamma distributions would often be closer to the sample moments; the lognormal is not a clear cut winner based on distance to the sample moments. for example, the Kolmogorov-Smirnov goodness-of-fit test smaller for the MLE lognormal fit than it is for the MLE Weibull fit. Hence, For the ball bearing data set, one should only conclude that the exponential & Pareto distributions are inferior in describing the distribution. For the Weibull distribution, the initial estimate of κ is close to the MLE, which was 2.10 (see section 4.2).

The Mathematica implementation is similar to the Weibull maximum likelihood problem: the major step is performed by a root-solver. Using Mathematica did lead us to a new discovery: although the kurtosis vs. skewness vs. coefficient of variation plot was only marginally useful in and of itself, the minimum Euclidean distance to the sample point provides us with some very useful exploratory results.

ACKNOWLEDGEMENT

This research was carried out during a Research Experience for Undergraduates sponsored by the National Science Foundation (grant number 310811) at The College of William & Mary during the summer of 1993. We thank the associate editor and referees for their helpful comments.

REFERENCES

[1] R.E. Barlow, F. Proschan, *Statistical Theory of Reliability and Life Testing: Probability Models*, To Begin With, Silver Springs, MD, 1981.

[2] D.R. Cox, D. Oakes, *Analysis of Survival Data*, Chapman and Hall, New York, 1984.

[3] J.F. Lawless, *Statistical Models and Methods for Lifetime Data*, John Wiley and Sons, New York, 1982.

[4] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, Second Edition, Addison–Wesley, 1991.

AUTHORS

Glen Hartless; Department of Statistics; Griffin–Floyd Hall, University of Florida; Gainesville, FL 32611 USA.

Glen Hartless received his BS in statistics from Virginia Tech, Blacksburg , VA in 1994. He is currently pursuing his Master's degree in statistics at the University of Florida. He is a student member of ASQC and ASA.

Lawrence Leemis; Department of Mathematics; College of William and Mary; Williamsburg, VA 23187-8795 USA.

Lawrence Leemis is a professor in the Mathematics Department at the College of William & Mary. He received his BS and MS degrees in Mathematics and his PhD in Industrial Engineering from Purdue University. He has served as an Associate Editor for the *IEEE Transactions on Reliability* and Book Review Editor for the *Journal of Quality Technology.*

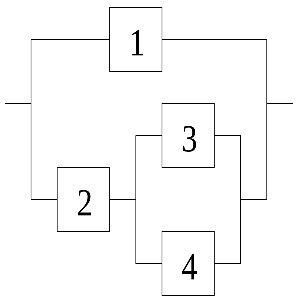

Figure 1: Four Component System

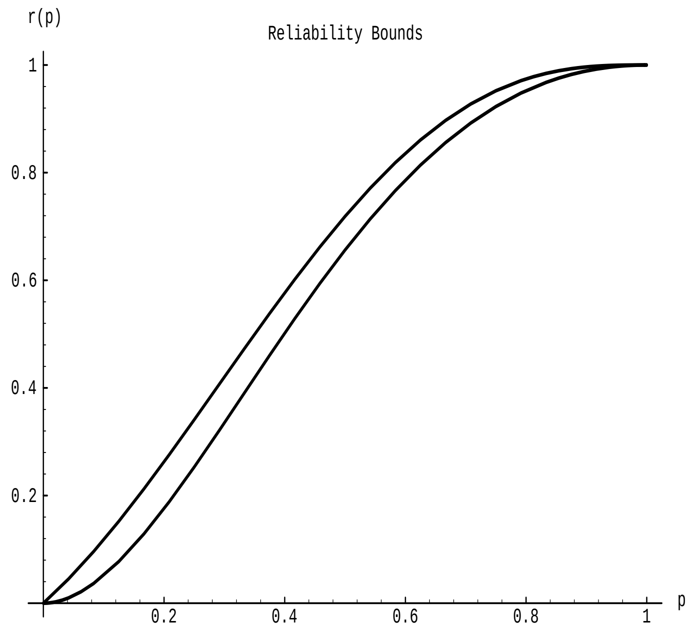

Figure 2: Reliability Bounds for a Four-Component System

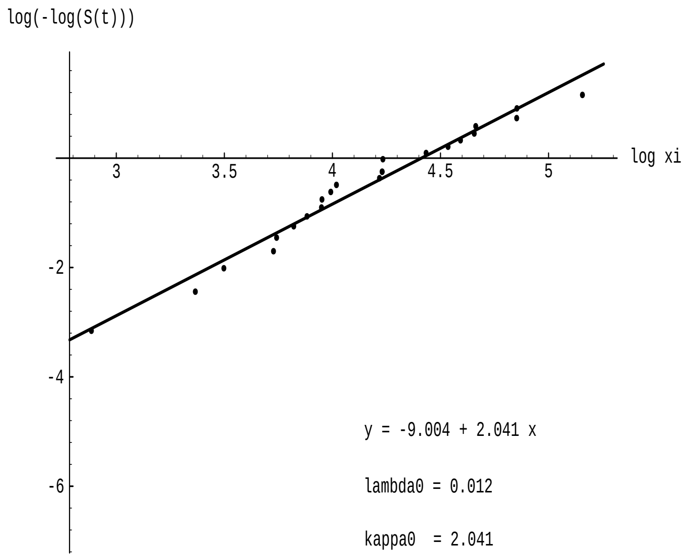

Figure 3: Weibull Plot (Ball Bearings, Weibull Fit)

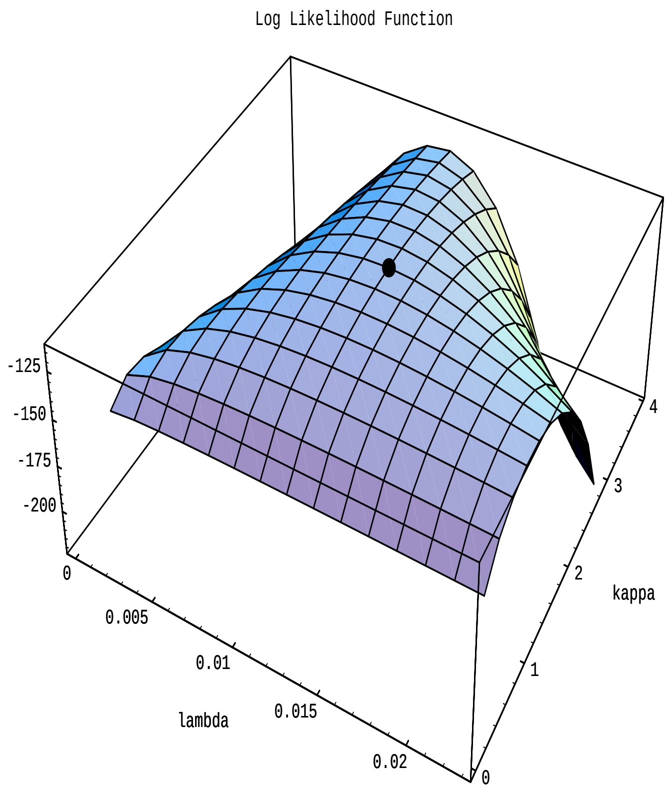

Figure 4: Log Likelihood Function with Maximum Value Identified (Ball Bearings, Weibull Fit)

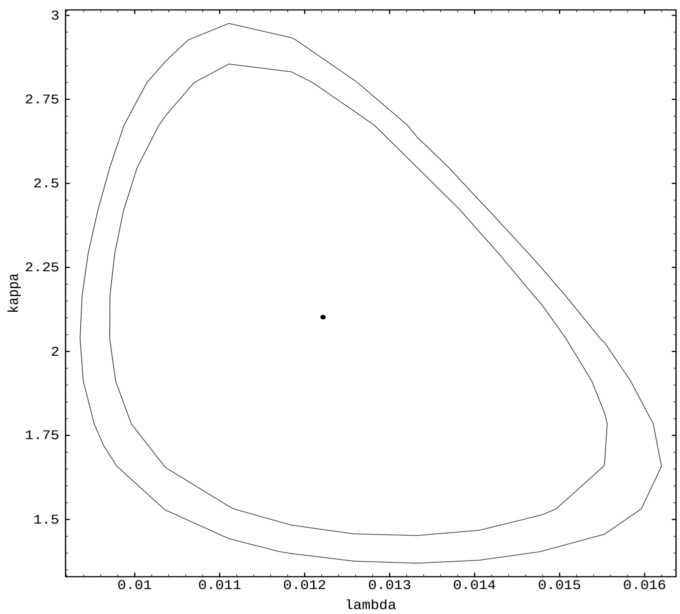

Figure 5: Confidence Region for MLEs ($\alpha = 0.10, 0.05$, Ball Bearings, Weibull Fit)

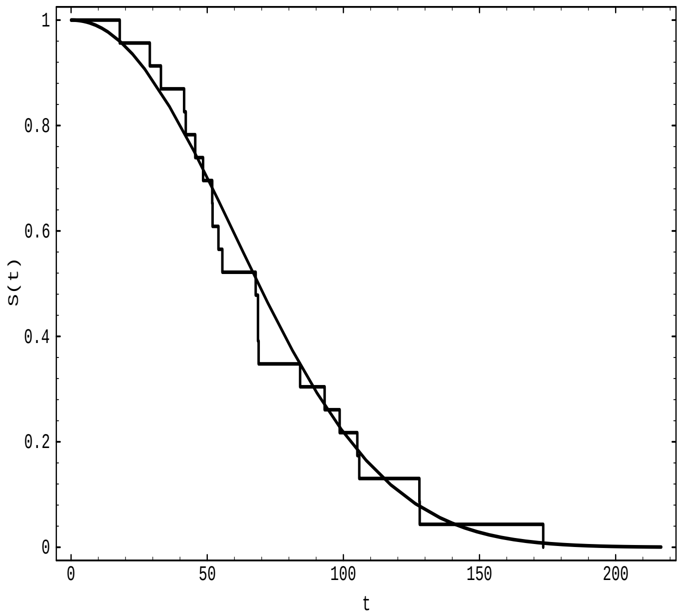

Figure 6: MLE Survivor Function vs. Kaplan–Meier Product-Limit Estimate (Ball Bearings, Weibull Fit)

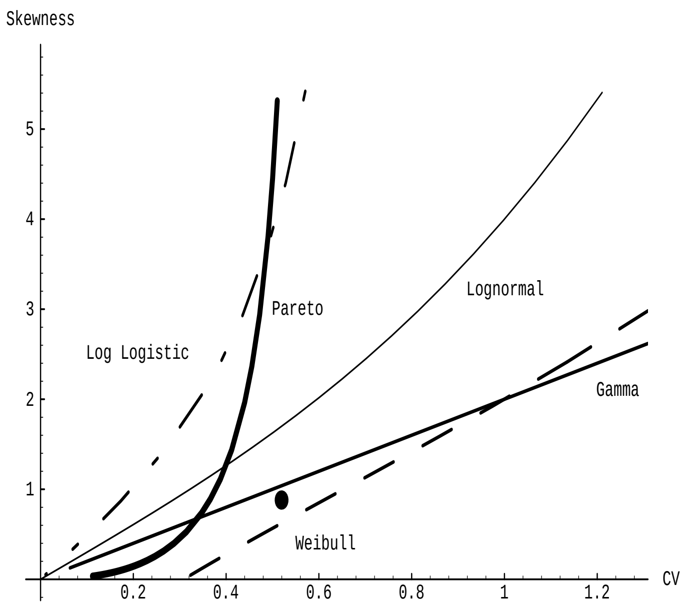

Figure 7: Skewness vs. Coefficient of Variation [Ball Bearings at $(0.52, 0.88)$]

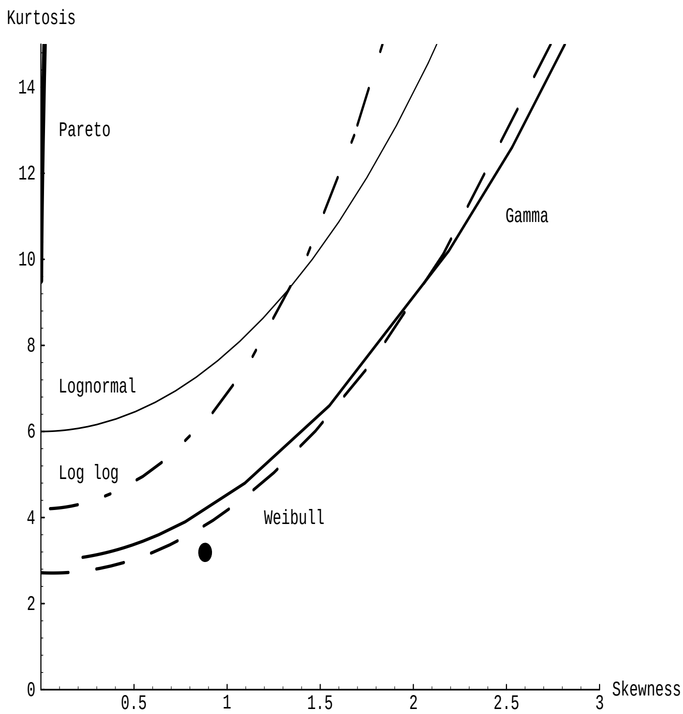

Figure 8: Kurtosis vs. Skewness [Ball Bearings at $(0.88, 3.19)$]

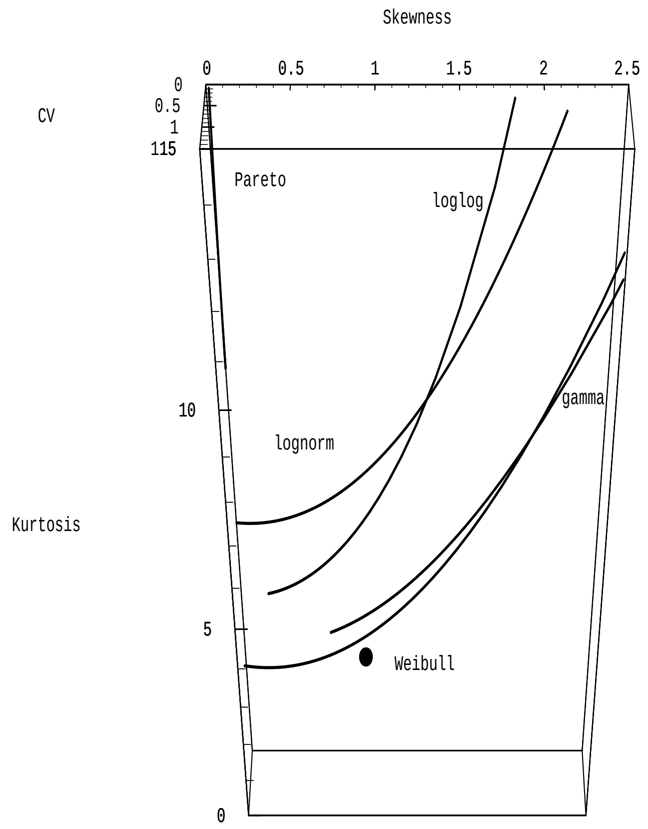

Figure 9: Kurtosis vs. Skewness vs. Coefficient of Variation [Ball Bearings at $(0.52, 0.88, 3.19)$]

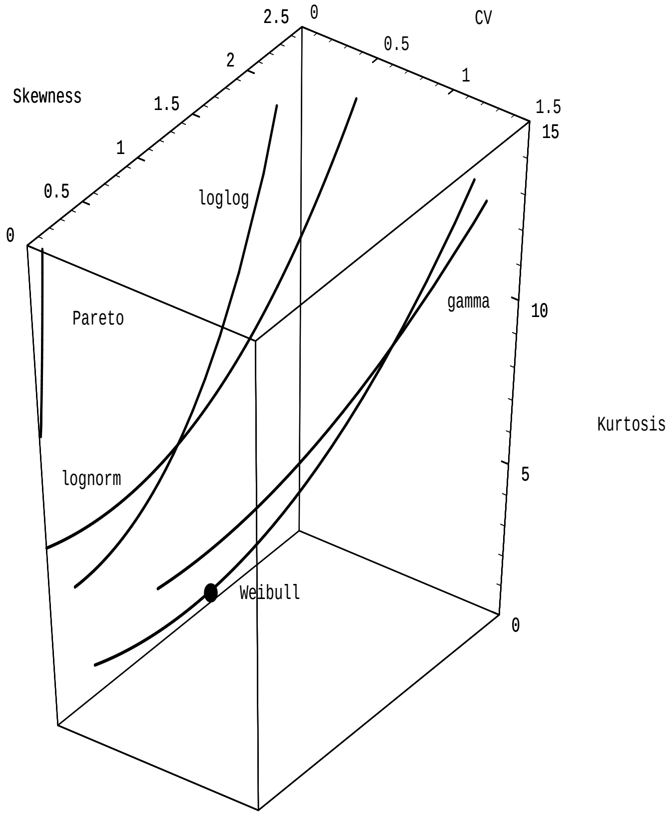

Figure 10: Kurtosis vs. Skewness vs. Coefficient of Variation, Rotated [Ball Bearings at $(0.52, 0.88, 3.19)$]