*Proceedings of the 2000 Winter Simulation Conference*
*J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.*

# INPUT MODELING USING A COMPUTER ALGEBRA SYSTEM

Diane L. Evans
Lawrence M. Leemis

Department of Mathematics
College of William & Mary
Williamsburg, VA 23187–8795, U.S.A.

## ABSTRACT

Input modeling that involves fitting standard univariate parametric probability distributions is typically performed using an input modeling package. These packages typically fit several distributions to a data set, then determine the distribution with the best fit by comparing goodness-of-fit statistics. But what if an appropriate input model is not included in one of these packages? The modeler must resort to deriving the appropriate estimators by hand for the appropriate input model. The purpose of this paper is to investigate the use of a prototype Maple-based probability language, known as APPL (A Probability Programming Language), for input modeling. This language allows an analyst to specify a standard or non-standard distribution for an input model, and have the derivations performed automatically. Input modeling serves as an excellent arena for illustrating the applicability and usefulness of APPL. Besides including pre-defined types for over 45 different continuous and discrete random variables and over 30 procedures for manipulating random variables (e.g., convolution, transformation), APPL contains input modeling procedures for parameter estimation, plotting empirical and fitted CDFs, and performing goodness-of-fit tests. Using examples, we illustrate its utility for input modeling.

## 1 PRELIMINARY EXAMPLES

There have been dozens of statistical languages developed over the years to relieve the computations associated with interactive or batch processing of data. APPL's data structures and algorithms were initially developed to accommodate probability problems, but may be used to solve input modeling problems as well. In order to illustrate the syntax and capability of APPL, we begin with some simple examples from probability theory in this section, then address some input modeling problems in the next section.

**Example 1.** Find the probability that the sum of eight independent and identically distributed U(0,1) random variables falls between $\frac{7}{2}$ and $\frac{11}{2}$. Letting $X_1, X_2, \ldots, X_8$ denote the U(0,1) random variables, the desired probability is

$$\Pr\left(\frac{7}{2} < \sum_{i=1}^{8} X_i < \frac{11}{2}\right).$$

The two standard methods for *approximating* the probability are the central limit theorem and Monte Carlo simulation. The central limit theorem approximation gives only one digit of accuracy for this particular problem. Monte Carlo simulation, on the other hand, converges to the exact value if a good random number generator is used, but requires custom coding and requires a 100-fold increase in computing time for each additional digit of accuracy. The APPL statements

```
n := 8;
X := UniformRV(0, 1);
Y := ConvolutionIID(X, n);
CDF(Y, 11 / 2) - CDF(Y, 7 / 2);
```

solve the problem exactly, yielding

$$\frac{3580151}{5160960}.$$

`ConvolutionIID` computes the exact distribution of the sum and stores the result in Y. This may be coded up more compactly as

```
Y:=ConvolutionIID(UniformRV(0,1),8);
CDF(Y, 11 / 2) - CDF(Y, 7 / 2);
```

**Example 2.** Let $X \sim triangular(1, 2, 3)$ and $Y \sim U(1, 2)$. If $X$ and $Y$ are independent, find the distribution of $V = XY$. The APPL code to solve this problem is

```
X := TriangularRV(1, 2, 3);
Y := UniformRV(1,2);
V := Product(X, Y);
```

which returns the probability density function of V as

$$f_V(v) = \begin{cases} v - \ln(v) - 1 & 1 < v \le 2 \\ -\frac{3}{2}v + 4\ln(v) + 4 - 5\ln(2) & 2 < v \le 3 \\ -\frac{1}{2}v + \ln(\frac{27}{32}v) + 1 & 3 < v \le 4 \\ \frac{1}{2}v - 3\ln(v) - 3 + \ln(216) & 4 < v < 6. \end{cases}$$

More complicated distributions than the triangular and uniform can be input in a similar manner.

**Example 3.** Let $X$ be a random variable associated with the Kolmogorov–Smirnov test statistic in the all-parameters-known case for sample size $n = 5$ under $H_0$. Similarly, let $Y$ be a Kolmogorov–Smirnov random variable (all parameters known) with $n = 3$. If $X$ and $Y$ are independent, find $\mathrm{Var}\,[\max\{X, Y\}]$. The APPL code to solve this problem is

```
X := KSRV(5);
Y := KSRV(3);
Z := Maximum(X, Y);
Variance(Z);
```

which yields $\frac{1036875145231938755837167 1}{6673923267539062500000000000}$ or approximately 0.0155362.

Since the base language for APPL is the symbolic language Maple, symbolic parameters can be accommodated, as illustrated in the next example.

**Example 4.** Let $X$ have the triangular distribution with minimum $a$, mode $b$, and maximum $c$. Find the CDF of $X$. The APPL code to determine the CDF is

```
X := TriangularRV(a, b, c);
CDF(X);
```

which yields

$$F(x) = \begin{cases} 0 & x \le a \\ \dfrac{(x-a)^2}{(c-a)(b-a)} & a < x \le b \\ 1 - \dfrac{(c-x)^2}{(c-a)(c-b)} & b < x \le c \\ 1 & x > c. \end{cases}$$

APPL is capable of computing the distribution of order statistics, as shown in the following two examples.

**Example 5.** Consider a sample of size $n = 7$ from a Weibull distribution with scale parameter $\lambda = \frac{1}{2}$ and shape parameter $\kappa = 2$ with PDF

$$f_X(x) = \frac{1}{2} x e^{-\frac{1}{4}x^2} \qquad x > 0.$$

Calculate the mean of the second order statistic. The mean of the second order statistic is

$$\frac{7}{6}\sqrt{6\pi} - \frac{6}{7}\sqrt{7\pi} \cong 1.0456613,$$

which is computed with the APPL commands

```
X := WeibullRV(1 / 2, 2);
Y := OrderStat(X, 7, 2);
Mean(Y);
```

Additionally, APPL is capable of performing operations on discrete random variables. The APPL data structure is similar to that for continuous random variables. There is a single format for continuous random variables, but two formats for discrete random variables.

**Example 6.** Define a geometric random variable $X$ with parameter $p = \frac{1}{4}$ to model the number of trials up to and including the first success, i.e., $f_X(x) = \frac{1}{4} \cdot \frac{3}{4}^{x-1}$, $x = 1, 2, \ldots$. Calculate the median of the maximum order statistic when $n = 5$ items are sampled *with replacement* from this geometric distribution. The APPL statements

```
X := GeometricRV(1 / 4);
Y := OrderStat(X, 5, 5);
IDF(Y, 0.5);
```

return the median of the distribution as 8.

A modeler is not limited to the built-in distributions introduced so far (e.g., UniformRV, TriangularRV, KSRV, WeibullRV). Any discrete or continuous random variable can be accommodated by using the data structure illustrated in the next example.

**Example 7.** Let the random variable $T$ have hazard function

$$h_T(t) = \begin{cases} \lambda & 0 < t < 1 \\ \lambda t & t \ge 1 \end{cases}$$

for $\lambda > 0$. Find the survivor function $S(t) = \Pr(T \ge t)$.

The APPL code requires inputting the hazard function for $T$ as a list of three sublists

```
assume(lambda > 0);
T := [[t -> lambda, t -> lambda * t],
      [0, 1, infinity],
      ["Continuous", "HF"]];
SF(T);
```

where the `assume` statement defines the parameter space. This yields the survivor function

$$S_T(t) = \begin{cases} e^{-\lambda t} & 0 < t < 1 \\ e^{-\lambda(t^2+1)/2} & t \ge 1. \end{cases}$$

**Example 8.** (Hogg and Craig 1995, page 287) Let $X_1$ and $X_2$ be iid observations drawn from a population with PDF

$$f(x) = \theta x^{\theta - 1} \qquad 0 < x < 1,$$

where $\theta > 0$. Test $H_0$: $\theta = 1$ versus $H_1$: $\theta > 1$ using the test statistic $X_1 X_2$ and the critical region $C = \{(X_1, X_2) | X_1 X_2 \geq 3/4\}$. Find the significance level $\alpha$ and power function for the test. The APPL code to compute the power function is

```
n := 2;
crit := 3 / 4;
assume(theta > 0);
X := [[x -> theta * x ^ (theta - 1)],
      [0, 1], ["Continuous", "PDF"]];
T := ProductIID(X, n);
power := SF(T, crit);
```

which yields

$$\Pr(\text{rejecting} \quad H_0 | \theta) = 1 - (3/4)^{\theta} + \theta (3/4)^{\theta} \ln(3/4).$$

The fact that the population distribution is non-standard indicates that `X` must be defined using the list of three sublists data structure shown above.

To compute the significance level of the test, the additional Maple statement

```
alpha := subs(theta = 1, power);
```

is required, yielding $\alpha = 1/4 + (3/4)\ln(3/4) \cong 0.0342$. To plot the power function requires the additional statement

```
plot(power, theta = 0 .. 4);
```

Obviously, this example can be generalized for different sample sizes, population distributions, and critical values with only minor modification.

**Example 9.** Consider the independent random variables $U_1 \sim U(0, 1)$ and $U_2 \sim U(0, 1)$. The Box–Muller algorithm for generating a single standard normal deviate $V$ can be coded in one line (Devroye 1996) as

$$V \leftarrow \sqrt{-2 \ln U_1} \cos(2\pi U_2),$$

where $U_1$ and $U_2$ are independent random numbers. Using the `Transform` (Glen, Leemis, and Drew 1997) and `Product` procedures together, one can determine the PDF of $V$. Due to the principle inverse difficulty with trigonometric functions, however, the transformation must be rewritten as

$$V \leftarrow \sqrt{-2 \ln U_1} \cos(\pi U_2)$$

before using `Transform` on the second factor in the expression. The APPL code

```
U1 := UniformRV(0, 1);
U2 := UniformRV(0, 1);
g1 := [[x -> ln(x)], [0, infinity]];
X1 := Transform(U1, g1);
g2 := [[x -> -2 * x],
       [-infinity, infinity]];
X2 := Transform(X1, g2);
g3 := [[x -> sqrt(x)],
       [0, infinity]];
X3 := Transform(X2, g3);
h1 := [[x -> Pi * x],
       [-infinity, infinity]];
Y1 := Transform(U2, h1);
h2 := [[x -> cos(x)],
       [-infinity, infinity]];
Y2 := Transform(Y1, h2);
V := Product(X3, Y2);
```

yields the following PDF for $V$

$$h(v) = \begin{cases} \dfrac{v}{\pi} \displaystyle\int_{-1}^{0} \dfrac{e^{-v^2/(2x^2)}}{x^2 \sqrt{1 - x^2}} \, dx & -\infty < v < 0 \\ \dfrac{v}{\pi} \displaystyle\int_{0}^{1} \dfrac{e^{-v^2/(2x^2)}}{x^2 \sqrt{1 - x^2}} \, dx & 0 < v < \infty. \end{cases}$$

While this form in not easily recognizable as the PDF for the normal distribution, it is mathematically equivalent to the more standard

$$h(v) = \frac{1}{\sqrt{2\pi}} e^{-v^2/2} \qquad -\infty < v < \infty.$$

We anticipate that future versions of Maple will be able to simplify these integrals.

**Example 10.** This example considers the use of the Kolmogorov–Smirnov test for assessing model adequacy (goodness of fit) for the prime modulus multiplicative linear congruential random number generator:

$$z_{i+1} = a z_i \mod m$$

for $i = 0, 1, \ldots$, where $z_0$ is a seed, $a = 7^5 = 16,807$, and $m = 2^{31} - 1 = 2,147,483,647$ (Park and Miller 1988). The random numbers generated are $z_1/m$, $z_2/m$, etc. If the seed $z_0 = 987,654,321$ is used, then the first five random numbers generated are

$$\frac{1,605,065,384}{2,147,483,647} \quad \frac{1,791,818,921}{2,147,483,647} \quad \frac{937,423,366}{2,147,483,647}$$
$$\frac{1,334,477,970}{2,147,483,647} \quad \frac{252,032,522}{2,147,483,647}$$

or, approximately

$$0.7474168 \qquad 0.8343807 \qquad 0.4365218$$
$$0.6214147 \qquad 0.1173618.$$

Since these five data values are being evaluated for their uniformity, there should be a reasonable match between their empirical cumulative distribution function and the cumulative distribution function for a U(0, 1) random variable. If we let the list `Sample` contain the five random numbers generated above, then the APPL statements required to plot these two functions over the interval (0, 1), shown in Figure 1, are

```
n := 5;
a := 7 ˆ 5;
seed := 987654321;
m := 2 ˆ 31 - 1;
Sample := [];
for j from 1 to n do
  seed := a * seed mod m:
  Sample := [op(Sample), seed / m]:
od;
U := UniformRV(0, 1)
PlotEmpVsFittedCDF(U,Sample,[],0,1);
```

The five parameters to the plotting function are the random variable whose CDF is to be plotted, the data values in a list, the parameters associated with the random variable (empty in this case of U(0, 1)), and the optional lower and upper horizontal plotting limits.

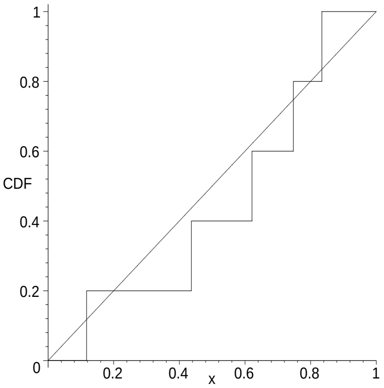

Figure 1: The Empirical CDF of `Sample` and the Theoretical U(0, 1) CDF

Let $F(x)$ be the hypothesized CDF and $F_5(x)$ be the empirical CDF. In order to determine the Kolmogorov–Smirnov test statistic,

$$D_5 = \sup_x |F(x) - F_5(x)|,$$

which measures the largest vertical distance between the two cumulative distribution functions, the following additional command must be issued

```
TestStat := KSTest(U, Sample, []);
```

The approximate value of the test statistic for the five random numbers is 0.2365, which occurs just to the left of the random number 0.4365.

Since large values of the test statistic indicate a poor fit and the cumulative distribution function $F_{D_5}(y)$ of the test statistic is (Drew, Glen and Leemis 2000)

$$\begin{array}{ll}
0 & y < \frac{1}{10} \\
\frac{24}{625}\,(10\,x-1)^5 & \frac{1}{10} \le y < \frac{1}{5} \\
-288\,x^4 + 240\,x^3 - \frac{1464}{25}\,x^2 + \frac{672}{125}\,x - \frac{96}{625} & \frac{1}{5} \le y < \frac{3}{10} \\
160\,x^5 - 240\,x^4 + \frac{424}{5}\,x^3 + 12\,x^2 - \frac{168}{25}\,x + \frac{336}{625} & \frac{3}{10} \le y < \frac{2}{5} \\
-20\,x^5 + 74\,x^4 - \frac{456}{5}\,x^3 + \frac{224}{5}\,x^2 - \frac{728}{125}\,x & \frac{2}{5} \le y < \frac{1}{2} \\
12\,x^5 - 6\,x^4 - \frac{56}{5}\,x^3 + \frac{24}{5}\,x^2 + \frac{522}{125}\,x - 1 & \frac{1}{2} \le y < \frac{3}{5} \\
-20\,y^6 + 32\,y^5 - \frac{185}{9}\,y^3 + \frac{175}{36}\,y^2 + \frac{3371}{648}\,y - 1 & \frac{1}{2} \le y < \frac{3}{5} \\
-8\,x^5 + 22\,x^4 - \frac{92}{5}\,x^3 + \frac{12}{25}\,x^2 + \frac{738}{125}\,x - 1 & \frac{3}{5} \le y < \frac{4}{5} \\
2\,x^5 - 10\,x^4 + 20\,x^3 - 20\,x^2 + 10\,x - 1 & \frac{4}{5} \le y < 1 \\
1 & y \ge 1,
\end{array}$$

the $p$-value for this particular test is found with the additional APPL statement

```
p := SF(KSRV(5), TestStat);
```

which yields $p \cong 0.8838$.

If this process is repeated for a total of 1000 groups of nonoverlapping consecutive sets of five random numbers, the empirical CDF of the Kolmogorov–Smirnov statistics should be close to the theoretical from APPL if the random number generator is valid. Figure 2 is a plot of the empirical CDF of the 1000 Kolmogorov–Smirnov statistics versus the theoretical Kolmogorov–Smirnov CDF with $n = 5$. The empirical CDF lies slightly above the theoretical. If this experiment were performed repeatedly, the empirical CDFs should fluctuate around the theoretical CDF.

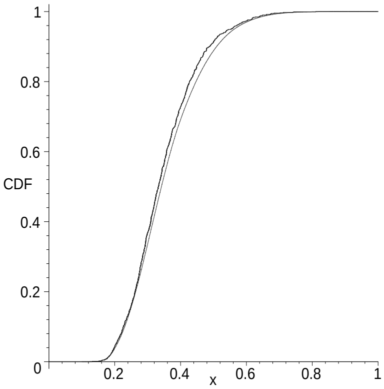

Figure 2: Empirical CDF of 1000 Kolmogorov–Smirnov Statistics and the Theoretical Kolmogorov–Smirnov CDF for $n = 5$

## 2 INPUT MODELING

Both APPL and Maple can easily be adapted for use in input modeling. This section gives several examples of cases where a symbolic language is of use in analyzing a data set.

**Example 11.** Model selection. One of the tools for selecting a suitable input model is a plot of the coefficient of variation ($\gamma = \sigma/\mu$) versus the skewness

$$\gamma_3 = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right].$$

After constructing this plot, the sample coefficient of variation and sample skewness can be plotted for a particular data set or data sets to determine an appropriate distribution for modeling the data.

The code that produces the plot in Figure 3 for the Weibull, gamma, log normal, and log logistic distributions uses the additional APPL procedures `CoefOfVar` and `Skewness`. The statements necessary to plot the gamma distribution's coefficient of variation versus skewness are shown below. The plots for the other distributions are calculated similarly. The Maple statement used to display all four plots in one graphic is also provided.

```
unassign('kappa');
lambda := 1;
X := GammaRV(lambda, kappa);
c := CoefOfVar(X);
s := Skewness(X);
GammaPlot := plot([c, s, kappa =
   0.5 .. 999],labels = [cv, skew]):
    .
    .
    .
plots[display]({GammaPlot,WeibullPlot,
     LogNormalPlot, LogLogisticPlot},
     scaling = unconstrained);
```

The `unassign` command in Maple is used to unassign any previous value given to an existing variable name, such as $\kappa$. Future `unassign` statements will be omitted for brevity.

**Example 12.** The following $n = 23$ ball bearing failure times (in $10^6$ revolutions) will be analyzed to determine a parametric input model in a discrete-event simulation. The failure times are (Lawless 1982, page 228)

| | | | | | |
|---|---|---|---|---|---|
| 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. | |

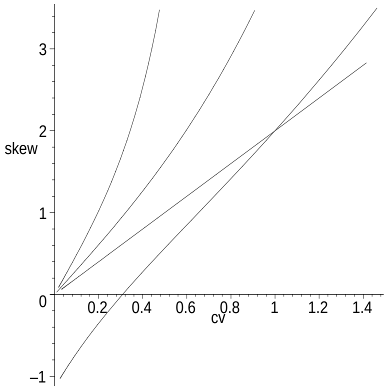

Figure 3: Coefficient of Variation, $\gamma$, Versus Skewness, $\gamma_3$, for the Gamma, Weibull, Log Normal, and Log Logistic Distributions

(The same principles that apply to the modeling of these ball bearing failure times also apply to the modeling of service times or stationary interarrival times for a queueing system.)

First, consider fitting an exponential distribution to this data set using maximum likelihood. The data set for the ball bearing failure times, `BallBearing`, is a pre-defined list in APPL. The APPL procedure `MLE` returns the maximum likelihood estimators as a list. Its arguments are the model, the data, and the parameters to be estimated. The APPL statements

```
X := ExponentialRV(lambda);
lamhat := MLE(X,BallBearing,[lambda]);
```

return $\hat{\lambda} \cong 0.0138$ as the maximum likelihood estimator. The additional APPL command

```
PlotEmpVsFittedCDF(X, BallBearing,
     [lambda = lamhat[1]], 0, 180);
```

where `lambda` = `lamhat[1]` assigns the value in the list `lamhat` to `lambda`, produces a plot of the empirical and fitted CDFs on one set of axes, as seen in Figure 4.

In order to assess the model adequacy, either a formal goodness-of-fit test can be performed, or goodness-of-fit statistics can be compared for competing models. The Kolmogorov–Smirnov test statistic, for example, can be computed with the additional APPL statement

```
KSTest(X, BallBearing,
    [lambda=lamhat[1]]);
```

which returns 0.3068, indicating a rather poor fit.

As an alternative, one might consider fitting the *reciprocal* of an exponential random variable to the ball bearing failure times, as suggested in the following example.

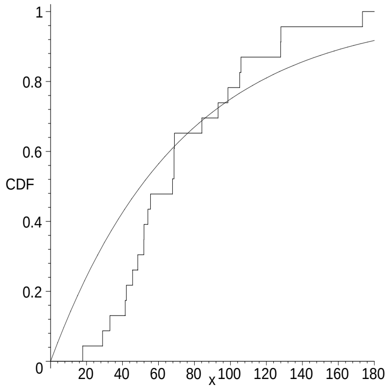

Figure 4: Empirical and Fitted Exponential Cumulative Distribution Functions for the Ball Bearing Data Set

**Example 13.** Fit the reciprocal of an exponential random variable to the ball bearing failure times in the previous example.

The APPL statements required to find the distribution of the reciprocal of an exponential random variable and find the MLE for the unknown parameter are

```
X := ExponentialRV(lambda);
g := [[x -> 1 / x], [0, infinity]];
Y := Transform(X, g);
lamhat := MLE(Y,BallBearing,[lambda]);
```

which derives the PDF of $Y$ to be

$$f_Y(y) = \frac{\lambda}{y^2} e^{-\lambda/y} \qquad\qquad y > 0$$

and calculates the MLE $\hat{\lambda} \cong 55.06$. The function `g` is used to find the distribution of $Y = g(X) = 1/X$.

As can be seen in Figure 5, the reciprocal of the exponential also provides a poor fit to the ball bearing data.

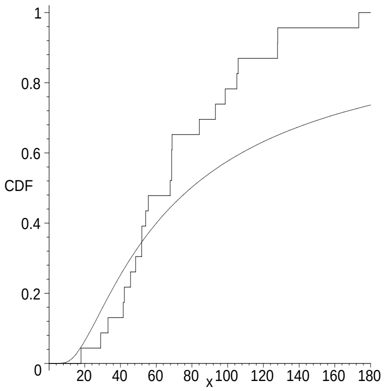

Figure 5: Empirical and Reciprocal Exponential Fitted Cumulative Distribution Functions for the Ball Bearing Data Set

Neither the exponential model nor its reciprocal are appropriate for modeling the failure times. It might be appropriate to consider two-parameter distributions as potential models, as shown in the next example.

**Example 14.** Fit the inverse Gaussian and Weibull distributions to the ball bearing failure times. Again using the APPL procedures `MLE` and `KSTest`,

```
X := InverseGaussianRV(lambda, mu);
hat :=MLE(X,BallBearing,[lambda,mu]);
KSValue := KSTest(X, BallBearing,
    [lambda = hat[1], mu = hat[2]]);
```

yields an improved fit with $\hat{\lambda} \cong 231.67$, $\hat{\mu} \cong 72.22$, and a Kolmogorov–Smirnov test statistic of 0.088. The procedure `MLE` is able to return the appropriate values because the maximum likelihood estimators are in closed form for this particular distribution. Unfortunately, the statements

```
Y := WeibullRV(lambda, kappa);
hat := MLE(Y, BallBearing,
        [lambda, kappa]);
```

fail to return the MLEs in APPL. The Maple numerical equation solving procedure `fsolve` is not clever enough to exploit some of the structure in the score vector that is necessary to find the MLEs. Therefore a special routine, `MLEWeibull`, has been written that computes MLEs for the Weibull distribution.

Besides the procedures `PlotEmpVsFittedCDF` and `KSTest`, fit can be assessed visually using a Q–Q or P–P plot (Law and Kelton 2000, pages 352–358). The APPL statements used to produce the Q–Q and P–P plots for the Weibull fit to the ball bearing failures displayed in Figures 6 and 7 are

```
QQPlot(Y, BallBearing,
  [lambda = hat[1], kappa = hat[2]]);
PPPlot(Y, BallBearing,
  [lambda = hat[1], kappa = hat[2]]);
```

To conclude the ball bearing data set examples, Table 1 summarizes the Kolmogorov–Smirnov test statistic values for various distributions that were fit to the data in APPL via maximum likelihood estimation.

Table 1: Kolmogorov–Smirnov Test Statistic Values for Various Distributions

| Model | Test statistic |
|---|---|
| Exponential | 0.307 |
| Reciprocal of Exponential | 0.306 |
| Weibull | 0.151 |
| Gamma | 0.123 |
| Arctangent | 0.094 |
| Log normal | 0.090 |
| Inverse Gaussian | 0.088 |

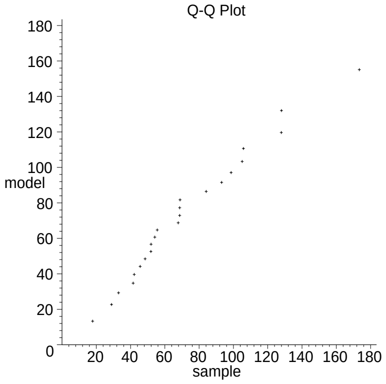

Figure 6: Q–Q Plot of Ball Bearing Data with Fitted Weibull Distribution

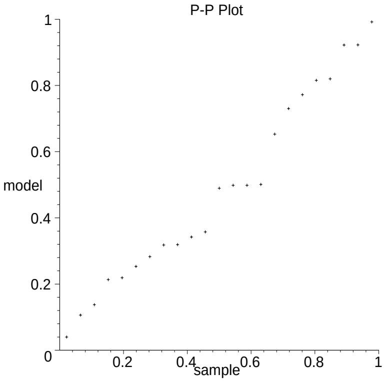

Figure 7: P–P Plot of Ball Bearing Data with Fitted Weibull Distribution

Another wrinkle that can present itself in input modeling is the presence of censoring. A right-censored data set, for example, often occurs in reliability and biostatistical applications. Examples likely to arise in discrete-event input modeling situations include machine failure times (when some machines have not yet failed) and the analysis of rare events.

**Example 15**. Consider the problem of determining an input model for the remission time for the treatment group in the study concerning the drug 6-MP (Gehan 1965). Letting an asterisk denote a right-censored observation, the remission times (in weeks) are

6 6 6 6* 7 9* 10 10* 11* 13 16
17* 19* 20* 22 23 25* 32* 32* 34* 35*.

Both `MP6` and `MP6Censor` are pre-defined lists in APPL. `MP6` is simply the 21 data values given above, and `MP6Censor` is the list

```
[1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1,
 0, 0, 0, 1, 1, 0, 0, 0, 0, 0]
```

where 0 represents a censored value and 1 represents an uncensored value. The statements used to determine the MLE for an exponential distribution are

```
X := ExponentialRV(lambda);
hat := MLE(X,MP6,[lambda],MP6Censor);
```

The code yields $\hat{\lambda} = \frac{9}{359}$. Similarly, the statement

```
hat := MLEWeibull(MP6, MP6Censor);
```

yields the MLE estimates $\hat{\lambda} \cong 0.03$ and $\hat{\kappa} \cong 1.35$ for the Weibull distribution. The Kaplan–Meier product-limit survivor function estimate for the `MP6` data set, along with the fitted Weibull survivor function, are plotted in Figure 8 using the additional APPL statements

```
Y := WeibullRV(lambda, kappa);
PlotEmpVsFittedSF(Y, MP6,
    [lambda = hat[1], kappa = hat[2]],
    MP6Censor, 0, 23);
```

The downward steps in the estimated survivor function occur only at observed remission times. The six parameters to the plotting function are the random variable whose SF is to be plotted, the data values in a list, the parameters associated with the random variable, the right-censoring vector in a list, and the lower and upper plotting limits. Note that the product-limit estimator cuts off after the largest observed remission time (Lawless 1982).

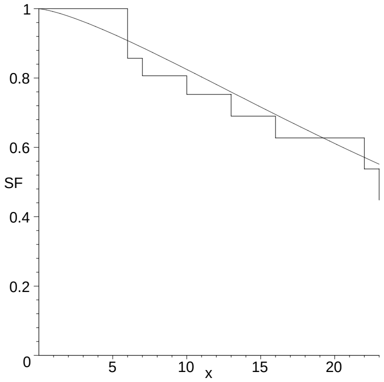

Figure 8: Product-Limit Survivor Function Estimate and Fitted Weibull Survivor Function for the 6-MP Treatment Group

All of the input modeling examples thus far have been limited to continuous data. The next example fits the geometric distribution as a model for daily demand at a vending machine.

**Example 16.** A vending machine has capacity for 24 cans of "Purple Passion" grape drink. The machine is restocked to capacity every day at noon. Restocking time is negligible. The last five days have produced the following Purple Passion sales:

$$14 \qquad 24 \qquad 18 \qquad 20 \qquad 24.$$

The *demand* for Purple Passion at this particular vending machine can be estimated from the data by treating the 24-can sales figures as *right-censored* demand observations. If demand has the geometric distribution, with probability function

$$f(t) = p(1-p)^t \qquad t = 0, 1, 2, \ldots$$

find the MLE for $\hat{p}$.

As discussed in the introductory section, many procedures, like `MLE`, are able to handle discrete distributions. Since the pre-defined geometric distribution in APPL is parameterized for $t = 1, 2, \ldots$, we need to define a geometric random variable with the different parameterization (used above) in the list of three sublists data structure. No new APPL commands are needed to compute the MLE for $\hat{p}$. The statements

```
X := [[x -> p * (1 - p) ^ x],
      [0 .. infinity],
      ["Discrete", "PDF"]];
PurplePass := [14, 24, 18, 20, 24];
PurplePassCensor := [1, 0, 1, 1, 0];
MLE(X, PurplePass, [p],
    PurplePassCensor);
```

yield $\hat{p} = \frac{3}{103}$. Model adequacy is not considered for this particular example.

All previous examples have considered time-independent observations. There are occasions when a series of event times may be time dependent, and a more complicated input model may be appropriate.

**Example 17.** Ignoring preventive maintenance, twelve odometer readings (from a certain model of car) associated with failures appearing over the first 100,000 miles are

| | | | | | |
|---|---|---|---|---|---|
| 12,942 | 28,489 | 65,561 | 78,254 | 83,639 | 85,603 |
| 88,143 | 91,809 | 92,360 | 94,078 | 98,231 | 99,900. |

Consider fitting a nonhomogeneous Poisson process to the above data set, where the ending time of the observation interval is assumed to be 100,000 miles. The data can be approximated by a power law process (i.e., the intensity function has the same parametric form as the hazard function for a Weibull random variable). The following APPL statements, including the additional procedure MLENHPP, return $\hat{\lambda} \cong 0.000026317$ and $\hat{\kappa} \cong 2.56800$:

```
CarFailures := [12942, 28489, 65561,
  78254, 83639, 85603, 88143, 91809,
  92360, 94078, 98231, 99900];
X := WeibullRV(lambda, kappa);
hat := MLENHPP(X, CarFailures,
               [lambda,kappa],100000);
```

The last argument in MLENHPP tells the procedure that the failures were observed over the interval [0, 100,000] miles. The additional APPL statement

```
PlotEmpVsFittedCIF(X, Sample,
  [lambda = hat[1], kappa = hat[2]],
  0, 100000);
```

produces a plot of the empirical cumulative intensity function and the fitted Weibull cumulative intensity function as shown in Figure 9.

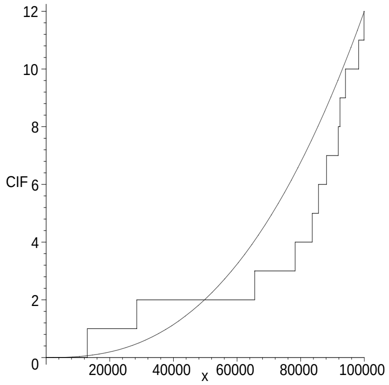

Figure 9: Cumulative Intensity Function Estimate and Fitted Weibull Cumulative Intensity Function for the `CarFailures` Data Set

Every example considered thus far has used maximum likelihood to estimate the unknown parameters. APPL includes the procedure `MOM` for computing the method of moments estimators.

**Example 18.** (Larsen and Marx, 2001, page 319) Hurricanes typically strike the eastern and southern coastal regions of the United States, although they occasionally sweep inland before completely dissipating. The U.S. Weather Bureau reported that during the period from 1900 to 1969 a total of 36 hurricanes moved as far as the Appalachian Mountains. The maximum 24-hour precipitation levels (measured in inches) recorded from those 36 storms during the time they were over the mountains are shown below and at the top of the following page.

A histogram of the data, which can be plotted in Maple, suggests that the random variable $X$, which is the maximum 24-hour precipitation, might be well approximated by the gamma distribution.

| | | | | | |
|---|---|---|---|---|---|
| 31.00 | 2.82 | 3.98 | 4.02 | 9.50 | 4.50 |
| 11.40 | 10.71 | 6.31 | 4.95 | 5.64 | 5.51 |

| | | | | | |
|---|---|---|---|---|---|
| 13.40 | 9.72 | 6.47 | 10.16 | 4.21 | 11.6 |
| 4.75 | 6.85 | 6.25 | 3.42 | 11.80 | 0.80 |
| 3.69 | 3.10 | 22.22 | 7.43 | 5.00 | 4.58 |
| 4.46 | 8.00 | 3.73 | 3.50 | 6.20 | 0.67. |

The following APPL code finds the method of moments estimates for the parameters $\lambda$ and $\kappa$, where `Hurricane` is the above data set pre-defined in APPL

```
X := GammaRV(lambda, kappa);
hat := MOM(X, Hurricane,
                [lambda, kappa]);
```

The resulting estimates for the parameters are $\hat{\lambda} = \frac{954000}{4252153} \cong 0.224$ and $\hat{\kappa} = \frac{6952275}{4252153} \cong 1.64$.

## 3 FURTHER WORK

Some ongoing work in the area of input modeling in APPL is described here. First, most distributions containing 3 or 4 unknown parameters (e.g., the Johnson distributions) are not going to have closed-form maximum likelihood estimators. Based on our experience with the Weibull distribution illustrated in Example 14, it will be necessary to write custom code for many of these distributions. This is precisely what is required from the batch and interactive software packages that perform input modeling. Fortunately, there is significant literature concerning the numerical methods required to arrive at these estimators.

Second, some distributions, such as the Erlang distribution, have both a discrete and a continuous parameter. In order to compute parameter estimates, it is necessary to prove results that will expedite their calculation. In using maximum likelihood on the Erlang, for example, it would not be possible to calculate the MLEs for the scale parameter for all shape parameters in the parameter space. Thus some results concerning the monotonicity of the likelihood function as the shape parameter varies are necessary to provide an algorithm for calculating the MLEs.

Third, some distributions have their unknown parameters as part of their support. Consider finding the MLEs for the triangular$(a, b, c)$ distribution for a sample size of $n = 2$. Without loss of generality, assume $x_1 < x_2$. Symmetry dictates that

$$\hat{b} = \frac{x_1 + x_2}{2}$$

and that $\hat{b} - \hat{a} = \hat{c} - \hat{b}$. Thus the problem of finding the MLE for $a$, for example, boils down to maximizing

$$f(x_1; a) = \frac{2(x_1 - a)}{(c - a)(b - a)} = \frac{x_1 - a}{(b - a)^2}.$$

Differentiating with respect to $a$ yields

$$\frac{\partial f}{\partial a} = \frac{-(b - a)^2 + 2(x_1 - a)(b - a)}{(b - a)^4}.$$

When the derivative is equated to zero and the resulting equation is solved for $a$, the MLE is

$$\hat{a} = 2x_1 - \hat{b}.$$

Likewise,

$$\hat{c} = 2x_2 - \hat{b}.$$

Moving to the case of $n = 3$ is more complicated since it is not clear whether the middle data value should have its likelihood function considered part of the left or the right support of the PDF. An algorithm must be developed in order to compute the MLEs for general $n$.

Fourth, an asymptotic confidence region for unknown parameters based on the likelihood ratio statistic can be determined by plotting the appropriate contour of the log likelihood function. Maple's symbolic and numeric abilities can be exploited to produce these plots for arbitrary distributions and data sets.

In conclusion, APPL is a platform which can be used for input modeling in an interactive, as opposed to a batch platform. Its ability to interface with probability theory presents some advantages for calculating exact probability measures. For further reading concerning the APPL software, see Glen, Leemis, and Evans (2000).

## ACKNOWLEDGMENTS

Diane Evans gratefully acknowledges support from the Clare Boothe Luce Foundation. The authors thank John Drew and Andy Glen for their contributions to the APPL language, and Steve Roberts for his helpful suggestions on the paper.

## REFERENCES

Devroye, L. 1996. Random variate generation in one line of code. *Proceedings of the 1996 Winter Simulation Conference*, ed. J. Charnes, D. Morrice, D. Brunner, J. Swain, 265–272. Institute of Electrical and Electronics Engineers, Coronado, California.

Drew, J. H., A. G. Glen, L. M. Leemis. 2000. Computing the cumulative distribution function of the Kolmogorov-Smirnov statistic. To appear, *Computational Statistics and Data Analysis*.

Gehan, E. A. 1965. A generalized Wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika* 52:203–223.

Glen, A. G., L. M. Leemis, and J. H. Drew. 1997. A generalized univariate change-of-variable transformation technique. *INFORMS Journal on Computing* 9:288–295.

Glen, A. G., L. M. Leemis, and D. L. Evans. 2000. APPL: A probability programming language. To appear, *The American Statistician*.

Hogg, R. V., and A. T. Craig. 1995. *Mathematical statistics*. 5th ed. Englewood Cliffs, New Jersey: Prentice–Hall.

Larsen, R. J., and M. J. Marx. 2001. *An introduction to mathematical statistics and it applications*, 3d ed. Englewood Cliffs, New Jersey: Prentice–Hall.

Law, A. M., and W. D. Kelton. 2000. *Simulation modeling and analysis*. 3d ed. New York: McGraw–Hill.

Lawless, J. F. 1982. *Statistical models and methods for lifetime data*, New York: John Wiley & Sons, Inc.

Park, S. K. and K. W. Miller. 1988. Random number generators: Good ones are hard to find. *Communications of the ACM* 31:1192–1201.

## AUTHOR BIOGRAPHIES

**DIANE L. EVANS** is a PhD student in Applied Science at The College of William & Mary. She received her BS and MA degrees in Mathematics from Ohio State University. She has also received an MS degree in Operations Research from the Mathematics Department at The College of William & Mary. She has taught in the Mathematics and Computer Science Department at Wittenberg University in Springfield, Ohio (1992–1994) and in the Mathematics Department at Virginia Wesleyan (1994–1998), and in the Mathematics Department at The College of William & Mary. Her research interests are in applied probability and operations research. She is a member of AMS, ASA, and INFORMS. Her email and web addresses are `<devans@math.wm.edu>` and `<www.math.wm.edu/∼devans>`.

**LAWRENCE M. LEEMIS** is a professor and chair of 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 also taught courses at Baylor University, The University of Oklahoma, and Purdue University. His consulting, short course, and research contract work includes contracts with AT&T, NASA/Langley Research Center, Delco Electronics, Department of Defense (Army, Navy), Air Logistic Command, ICASE, Komag, Federal Aviation Administration, Tinker Air Force Base, Woodmizer, Magnetic Peripherals, and Argonne National Laboratory. His research and teaching interests are in reliability and simulation. He is a member of ASA, IIE, and INFORMS. His email and web addresses are `<leemis@math.wm.edu>` and `<www.math.wm.edu/∼leemis>`.