When Regression Fails:
Combining JMP® Techniques to Handle Nonconforming Data

Cheryl E. Johnson, Data Analytics & Modeling
Shared Operations Services & Technology, Bank of America, Charlotte, NC 28255

Introduction

A technology team at Bank of America wished to use historical data to predict the performance of new projects based on the combination of the applications being tested for programming bugs. This team is tasked with designing and running test scripts to identify bugs so that they can be corrected prior to putting any applications with updates or new functionality back into production. Data Analytics & Modeling supports this team and was asked to develop the predictive model.

The objective of the analysis was two-fold: (1) initially estimate count and type of bugs and the timeframe in which they would likely to be detected, and (2) predict whether a project already underway could be completed on-time based on issues which had already been identified and the expected length of time to resolve them.

Methods Abandoned – Why Regression Failed

After a cursory examination of the data, traditional regression techniques were not deemed viable for three distinct reasons:

1. Potential predictors were all nominal variables with multiple levels – the technology team managed 40+ applications; other potential predictors had at least four levels each.
2. All predictors were very unevenly distributed amongst their levels – 69% of bugs came from just 10% of the applications due to the relative complexity of these applications and the frequency of upgrades. This uneven distribution of data would preclude the use of interaction terms in a regression model, despite their known influence.
3. Dependent variables were time (in cycles) to detect and to correct a bug, respectively. Neither was expected to be a simple linear combination of the predictor variables.

Methods Used – Survival, Partition and Distribution Analysis

The model for time to correct a bug was addressed first. Since this question can be viewed as a form of survival analysis (i.e., how long will the problem persist before it is resolved), exploratory analysis started with the **JMP® *Reliability and Survival*** platform. The distribution for the sample population of approximately 2800 bugs was heavily skewed. The overall average close time was approximately 2 cycles, but if an issue was still open after the first cycle, it would take an average additional 3 cycles to close.

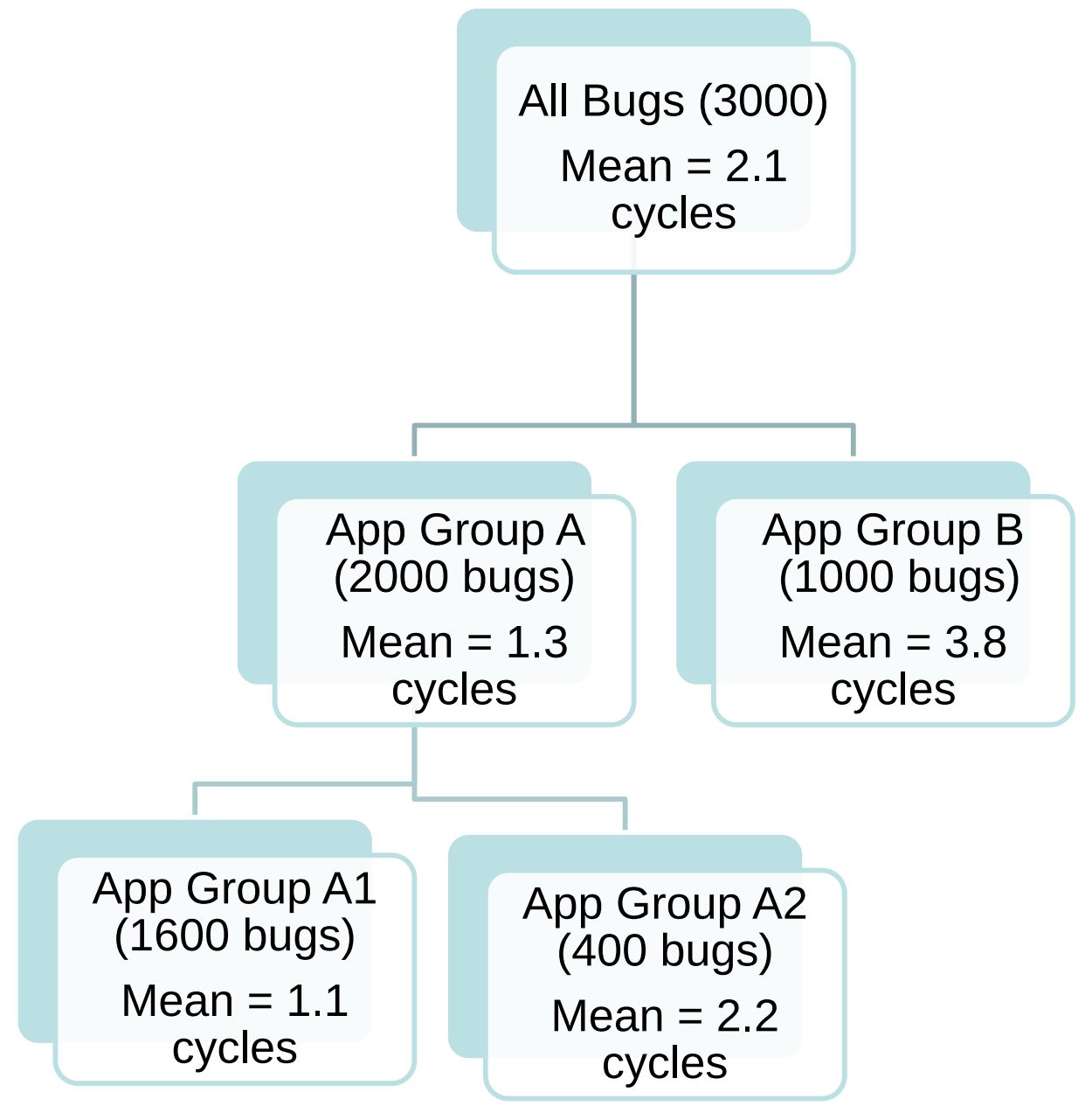
This behavior suggested fitting the data to a Weibull distribution, frequently used in survival modeling. However, the response clearly varied based on the application being tested and several other key predictors. Therefore, it was not viable to build a prediction model based on a single Weibull distribution.

Because of its flexibility and ability to handle interactions with unevenly distributed data, the ***Partition Model*** platform (Figure 1) was used to segment the data into appropriate "peer groups". An initial analysis, based strictly on the "best" splits, resulted in some odd results due to the significant gaps in the data. A more structured approach was subsequently used based on the available candidate report statistics and subject matter expertise.

The key to success was not allowing previous predictors to be reintroduced after creating a partition using a new predictor. This forced a "hierarchical" split resulting in 18 separate "peer" groups based on 3 predictor variables. (See Figure 1 below).

After saving the partition leaf numbers to the data set, the ***Distribution*** platform was executed using the leaf number as a BY variable. The parameter estimates and their confidence intervals were saved to a combined data table for further analysis. As part of the model diagnostics, confidence intervals for the Weibull coefficients were compared for overlap to ensure that no spurious splits had been made during the partition process.

Partition and distribution analyses were also used to model the time to detect a bug. A different set of predictors resulted from this partition analysis, but the Weibull distribution was again the best fit distribution. The two sets of partition means and Weibull coefficients were used in computations in a separate custom analysis tool. (See section at right).


All Bugs (3000)
Mean = 2.1 cycles

App Group A (2000 bugs)
Mean = 1.3 cycles

App Group B (1000 bugs)
Mean = 3.8 cycles

App Group A1 (1600 bugs)
Mean = 1.1 cycles

App Group A2 (400 bugs)
Mean = 2.2 cycles

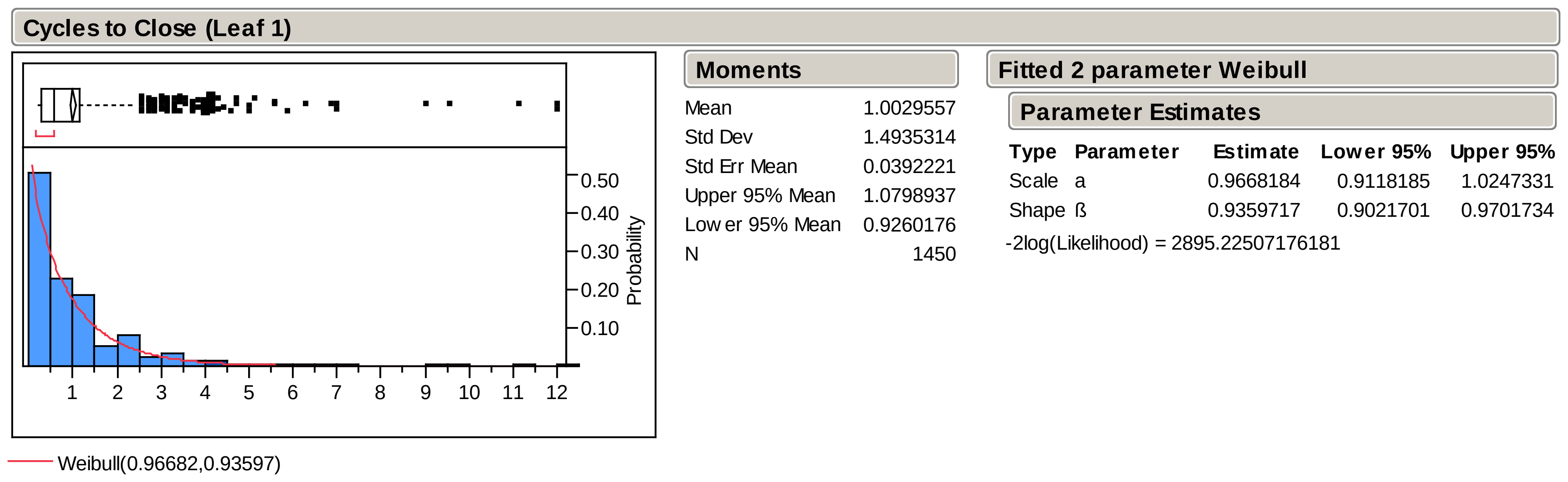
Final analysis involved 17 steps (18 groups)

Application plus two additional predictors were used as predictor variables

Figure 1. Depiction of First Two Steps of JMP ® Partition Analysis for # Cycles to Fix a Bug
(NOTE: All statistics and values shown in the figure are for illustrative purposes ONLY and are not based on actual data)

Results

Since each testing initiative is a unique combination of applications, software modifications and test scripts, a highly precise model was never anticipated. Rather, the desired outcome was the ability to distinguish between initiatives where the bugs could be resolved quickly and easily (Figure 2) vs. an initiative where bug resolution would be expected to drag out (Figure 3). Similarly, it was necessary to estimate whether most bugs could be detected within the first few cycles of an initiative vs. a more extended period. By incorporating the results of these two complementary models into the custom analysis tool described below, the dual objectives of the analysis were achieved.


Cycles to Close (Leaf 1)

Moments

Mean	1.0029557
Std Dev	1.4935314
Std Err Mean	0.0392221
Upper 95% Mean	1.0798937
Low er 95% Mean	0.9260176
N	1450

Fitted 2 parameter Weibull

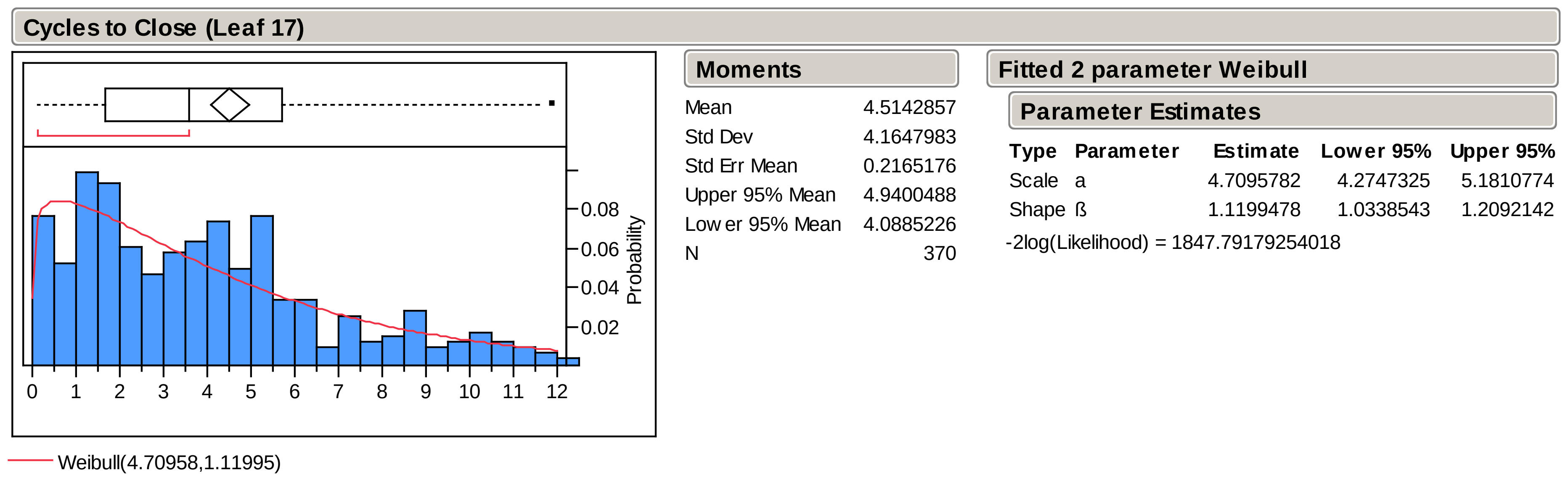
Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Scale	a	0.9668184	0.9118185	1.0247331
Shape	ß	0.9359717	0.9021701	0.9701734

-2log(Likelihood) = 2895.22507176181

Weibull(0.96682,0.93597)

Figure 2. Distribution Analysis for Time in Cycles to Close (Leaf 1) with Weibull Distribution Overlay (in Red)


Cycles to Close (Leaf 17)

Moments

Mean	4.5142857
Std Dev	4.1647983
Std Err Mean	0.2165176
Upper 95% Mean	4.9400488
Low er 95% Mean	4.0885226
N	370

Fitted 2 parameter Weibull

Parameter Estimates

Type	Parameter	Estimate	Lower 95%	Upper 95%
Scale	a	4.7095782	4.2747325	5.1810774
Shape	ß	1.1199478	1.0338543	1.2092142

-2log(Likelihood) = 1847.79179254018

Weibull(4.70958,1.11995)

Figure 3. Distribution Analysis for Time in Cycles to Close (Leaf 17) with Weibull Distribution Overlay (in Red)

How the Weibull Models were Used in a Custom Analysis Tool to Make Predictions

- The *initial* estimate for the expected number and type of bugs is based on the applications being tested and their historical averages and distributions by type.
- To assign when such a *future* bug would be expected to occur in the process, a random number between 0 and 1 is generated for each period.
  - The random number is normalized by dividing by the expected number of bugs for that application.
  - If the normalized random number is less than the probability predicted by the appropriate Weibull distribution, then a bug is assigned for that period; otherwise no bug is assigned.
- Actual "detected on" dates are used for bugs which have already been identified.
- The time to close a predicted *future* bug is estimated based on the mean closure time for the partition group into which it falls.
- The estimated time to close a *currently open* bug is estimated as ($t + L$) based the conditional survival probability[1]:

  $$[\ S(t) - S(t + L)]\ /\ S(t) \qquad \text{where}$$

  - The survival functions $S(t)$ *and* $S(t + L)$ are evaluated using the appropriate Weibull distribution
  - t is the current time period
  - L is additional time needed to ensure that the estimated conditional probability is barely greater than a specified threshold probability limit (e.g., 0.95 or 0.99)
- The closed, open, and predicted future bugs, combined with their actual or projected closure times, are used to generate a graphical view of expected performance by application throughout the entire project period. This can be used to (1) estimate resource needs while still in the planning stages of a project or (2) give early warning of the possibility of missing a critical deadline for those projects already underway.

References

1. Wang DE, Ries LA. On the Estimation of Survival. *Semin Surg Oncol.* 1994;10:2–6.

Acknowledgement

The author wishes to acknowledge teammates Greg Gunn and Ben Porter who collaborated on this project and developed the custom analysis tool.