Journal of Cost Analysis and Parametrics, 5:25–51, 2012

ISSN: 1941-658X print
DOI: 10.1080/1941658X.2012.682935

www.iceaaonline.com

Prediction Bounds for General-Error-Regression Cost-Estimating Relationships

STEPHEN A. BOOK

MCR, LLC, El Segundo, California

Introduction

Estimating the cost of a system under development is essentially trying to predict the future, which means that any such estimate contains uncertainty. When estimating using a cost-estimating relationship (CER), a portion of this uncertainty arises from the possibility that the cost-estimating form to which regression analysis is applied may be the incorrect one. That is, the data may have been fit to a linear form, but some curvilinear relationship may more appropriately model the data. Assuming the algebraic model being used is the correct one, the CER's uncertainty is described by its standard error of the estimate (SEE), which is basically the standard deviation of errors made (residuals) in applying that CER to estimate the (known) costs of the systems comprising the historical database. The SEE depends primarily on the extent to which those (known) costs fit the CER that purports to model them. Finally, additional uncertainty associated with a specific CER arises from the location of the particular cost-driver value ($\boldsymbol{x}$) within or without the range of cost-driver values for programs comprising the historical cost database. For example, if $\boldsymbol{x}$ were located near the center of the range of its historical values, the CER would provide a more precise measure of the element's cost than if $\boldsymbol{x}$ were located toward the edges or even outside the data range. The total uncertainty of CER-based estimates is a combination of all sources of uncertainty.

The first kind of uncertainty mentioned, which questions the particular CER shape involved, cannot be measured without redoing the regression analysis for a wide variety of algebraic and other kinds of CER forms. Once we have decided upon a definite CER form, the SEE, represented by only one number characteristic of the CER, is fairly easy to measure for any CER shape or error model using known algebraic formulas. The second kind of uncertainty associated with a specific CER, which assesses both the CER itself and the value of the cost-driving parameter, is more complicated, and the way to account for it is completely understood *only* in the case of classical linear regression, i.e., ordinary least squares (OLS). As a result, explicit formulas exist for "prediction intervals" that bound cost estimates based on CERs that have been derived by applying OLS to historical cost data.

For CERs, even linear ones, derived by other statistical methods, there appears to be no general method of solution described in the theoretical statistical literature. This report illustrates the application of bootstrap statistical sampling, a 34-year-old statistical process (Casella, 2003), to the problem of estimating prediction bounds for multiplicative-error and other CERs derived by non-OLS methods. After the bootstrap method is shown to be capable of yielding prediction bounds that approximate the known OLS bounds fairly

TABLE 1 A One-cost-driver dataset

Project	Actual cost	Cost-driver value
Project 1	y_1	x_1
Project 2	y_2	x_2
•	•	•
•	•	•
•	•	•
Project n	y_n	x_n

well, it is applied analogously to non-OLS-derived CERs. Although statistical sampling can yield only approximations to the "true" prediction bounds, the bootstrap technique appears to be a practical and theoretically credible method of approaching this currently unsolved estimating problem.

Prediction Bounds

The discussion of prediction bounds begins with a short summary of the state of the art in CER-based cost-estimating, excluding some more advanced atypical techniques that have been and will continue to be discussed elsewhere. To keep things as simple as possible, the discussion is restricted to the case of one-cost-driver (called "univariate") CERs.

OLS, also known as "classical linear regression," is a technique that appeared in the 18th century or earlier and was formally published by Gauss (1809, 1821/1823) in the early 19th century. An OLS CER models cost as an additive-error linear function of one or more cost drivers under a number of explicit mathematical conditions that will be discussed in detail below. The OLS estimating problem has been completely solved over the course of the last two centuries, even though new facts about it are still being discovered. Explicit algebraic formulas exist for the coefficients of the linear model, the standard error, the coefficient of determination (R^2), hypothesis tests for significance of the coefficients, and prediction bounds, which in the one-cost-driver situation are upper and lower bounds on the cost ("prediction intervals") for any value of the cost driver at any level of confidence. The width of the prediction interval depends on both the CER's SEE and the location of the cost-driver value of interest. Historical cost and technical data listed in a framework, such as that in Table 1, with x representing the cost driver, y the cost corresponding to it, and n the total number of data points, are the basis of all CER-related calculations. The formula for the prediction bounds is then

$$\hat{Y} \pm t_{\alpha/2,n-2} * SEE \sqrt{1 + \frac{1}{n} + \frac{(x-\bar{x})^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2}}. \quad (1)$$

Here, $\hat{Y}$ is the CER-based estimate of the cost at cost-driver value x, $\bar{x}$ is the mean (average) of the n cost-driver values, and $t_{\alpha/2,n-2}$ is the value of the t distribution for a two-sided 100 α% confidence interval, where α is a number between 0 and 1, inclusive.

Figure 1 illustrates an OLS CER bounded by hyperbola-shaped upper and lower bounds (the dashed lines, respectively). Notice that the bounds widen as the cost-driver

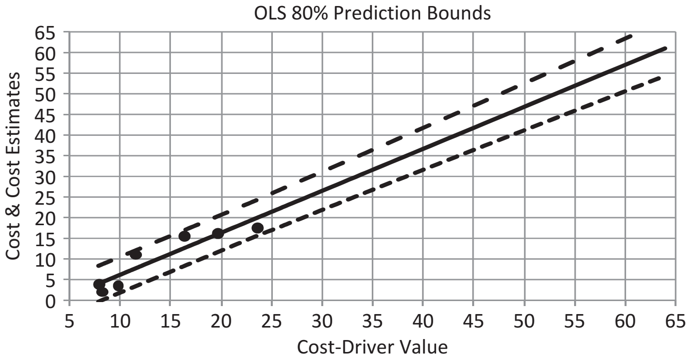

FIGURE 1 Prediction bounds for an OLS CER.

value moves away from the center of the dataset. In fact, the center of the dataset is the point $(\bar{x}, \bar{y})$, which is on the CER line (the solid line) in the OLS case.

A number of special non-linear algebraic forms can be fit using OLS techniques, but closed-form solutions (Equation (1) is an illustration) for SEE, R^2, and prediction bounds are problematical or non-existent. Some non-linear forms can be made OLS-solvable by an algebraic transformation (usually logarithmic), but the traditional OLS formula for R^2 is not applicable to non-OLS CER-based estimates; see Book and Young (2008) and Hu (2010) for details. Care has to be taken in such cases that SEE and other CER quality metrics are calculated in terms of x and y, rather than $\log(x)$ and $\log(y)$ or whatever transform space is used. Polynomial forms, such as quadratic, cubic, etc., can be solved using multiple-linear OLS methods. Closed-form expressions for prediction bounds do not appear to exist in any of these non-OLS situations.

General non-linear CER forms allow the modeling of cost using any error form, additive or multiplicative, and any algebraic form, linear or non-linear. For details, see Wedderburn (1974), Nguyen et al. (1994, 2004), Jørgensen (1997), Book and Young (1997), Book and Lao (1998), and Goldberg and Tuow (2003). In all of these cases, the standard error can be calculated, as well as useful stand-ins for other quality metrics, such as R^2, but the prediction-interval problem does not appear to have been solved for CERs derived by general-regression methods. One technique of approximating a solution to that problem is the subject of this article.

The Bootstrap Philosophy

While waiting for the “exact” theoretical solution of the prediction-interval problem to be found for non-OLS CERs (which, if history is a guide, could take decades), it would be useful to have available a practical “ad hoc” method that can be applied to generate prediction bounds in any particular case. “Bootstrap” statistical sampling appears to be an appropriate technique to consider. The bootstrap method of error estimation was introduced by Efron in 1977 (Efron, 1979) and has a 34-year history behind it; see Efron (2003) for a later discussion from his point of view. The bootstrap method is “distribution free,” so it does *not* require common (but difficult to verify) distributional assumptions, e.g., normal or lognormal error distributions or homoscedasticity. It works with additive- and multiplicative-error models and all algebraic equation forms.

The bootstrap philosophy parallels the philosophy of OLS and general-error regression in assuming the following scenario:

1. the true relationship between $\boldsymbol{y}$ (cost) and $\boldsymbol{x}$ (cost-driving parameter) is exactly the algebraic relationship that is being modeled;
2. the $\boldsymbol{x}$ values in the historical database are known precisely, but the "actual" $\boldsymbol{y}$ values are known accurately only to within some statistical error distribution;
3. the error distribution of $\boldsymbol{y}$ depends on how well its algebraic relationship with $\boldsymbol{x}$ accounts for the various influences on $\boldsymbol{y}$; and
4. the set of "residuals" (by which, for bootstrapping purposes, means the "actual" minus the "estimated" costs) represents the distribution of error in the actual $\boldsymbol{y}$ values.

The combined assumptions of the bootstrap philosophy imply that which residual happens to be matched with which particular $\boldsymbol{x}$ value is merely a matter of chance. This means that the residuals are assumed to be randomly (in this case, equally likely) selected from an (unknown) error distribution, of which the residuals themselves are the only manifestation. The conclusion that follows from this assumption is that had data been collected in a different way or at a different time or from different sources, any one of the residuals might have been obtained for any of the $\boldsymbol{x}$ values. Another way to look at the error distribution is to consider the residual corresponding to any particular data point as a random number drawn from the population comprised of all the residuals.

It is at this point that the bootstrap model departs from OLS. In OLS, the error distribution is postulated to be the normal distribution with mean 0 and standard deviation $\boldsymbol{\sigma}$, the numerical value of σ being the same for all points in the database. In particular, OLS assumes that the residual associated with any particular data point is a random number drawn from the normal distribution with mean 0 and standard deviation $\boldsymbol{\sigma}$**.** Bootstrapping does not require the normal distribution—its error distribution is defined solely by the set of residuals.

The most significant consequence of the way the bootstrap error distribution is defined and, as must be emphasized, the defining characteristic that makes bootstrapping a successful statistical method, is that by selecting for each cost-driver value $\boldsymbol{x}$ a random number from the set of residuals, a set of $\boldsymbol{y}$ values can be constructed that could very well have been the "actual" $\boldsymbol{y}$ values resulting from conducting the same data collection effort under different circumstances.

What Is Bootstrap Sampling?

Bootstrap sampling is a "resampling" method, where several (the more the better) random samples are taken, not from a probability distribution, such as the triangular, normal, or lognormal, but from the set of residuals that result from the derivation of a CER. The residuals are calculated from the *actual* database from which the CER was derived, *not* from an *assumed* probability distribution. As mentioned above, bootstrap theory assumes that each of the $\boldsymbol{n}$ residuals ($\boldsymbol{n}$ = number of data points) has probability $\mathbf{1}/\boldsymbol{n}$ of being *the* residual associated with any given $\boldsymbol{x}$ value. This assumption forces us to a process called "sampling with replacement," because a residual's association with one $\boldsymbol{x}$ value does not preclude its association with another $\boldsymbol{x}$ value in the same dataset. A good way to understand this is to view a probability distribution or residuals as a collection of numbers, like those painted on billiard balls, such that when one is drawn out of a tub, its number is recorded and then it is put back in the tub before the next number is drawn. Therefore, the same number could appear in the random sample more than once. An analog of this is the process of drawing a random number from a normal distribution. Although the normal distribution is a collection of infinitely many numbers, it is possible, at least in theory, to draw the same

TABLE 2 Database OLS setup for bootstrap sampling

x Values (cost driver)	y Values (actual costs)	Predicted y values (cost estimates)	Residuals = Actuals−Estimates
7.9	3.595	3.699	−0.104
8.2	1.900	4.005	−2.105
9.8	3.300	5.635	−2.335
11.5	10.900	7.367	3.533
16.4	15.434	12.358	3.076
19.7	16.074	15.720	0.354
23.6	17.274	19.693	−2.419

Note. CER derived from x and actual y values is $y = a + bx$, where $a = -4.348$ and $b = 1.0187$.

number more than once. Admittedly, in the case of the normal distribution, the probability is essentially zero.

Only the probability $\mathbf{1}/\boldsymbol{n}$ is hard-wired into the process. After a set of sample residuals has been randomly generated by the sampling-with-replacement process, a new dataset, called "the bootstrap sample," that could have been under other circumstances the "actual" dataset, is calculated from each set of $\boldsymbol{n}$ sample residuals. Next, a "bootstrap CER" that could have been the "real" CER (had the bootstrap sample been the real dataset) is calculated from each bootstrap sample. This process is repeated many times, and many sets of $\boldsymbol{n}$ sample residuals are generated, leading to many bootstrap samples and many bootstrap CERs.

To illustrate how this process works in practice, consider the dataset in Table 2, where $\boldsymbol{x}$ is the radar diameter in feet, and $\boldsymbol{y}$ is the cost in thousands of fixed-year dollars from which an OLS CER is derived:

The next step is to draw 255 random samples of size $\boldsymbol{n} = \mathbf{7}$ without replacement from the set of $\boldsymbol{n} = \mathbf{7}$ residuals listed in the far-right column of Table 2. Why 255 you ask? Well, why not? Microsoft Excel has 256 columns, and one is needed for the list of cost-driver values. Of course, the table could be transposed and be able to draw 1,048,575 residual samples and still have one row left over for the cost-driver values, but this is only an example. The first few of the 255 residual samples are displayed in Table 3.

TABLE 3 The first few of 255 random samples of residuals

x Values (cost driver)	Residual samples	#1	#2	#3	#4	#5	#6	...
7.9	First residual	−0.104	−2.105	−2.335	3.533	3.076	−2.419	...
8.2	Second residual	0.354	−2.419	−0.104	0.354	−2.335	−2.419	...
9.8	Third residual	−0.104	3.533	3.533	−2.105	−2.335	0.354	...
11.5	Fourth residual	−2.419	3.533	−2.419	−2.419	−2.105	−2.419	...
16.4	Fifth residual	−2.105	−2.105	3.533	−2.105	0.354	0.354	...
19.7	Sixth residual	−2.105	3.533	−2.105	−2.105	−2.419	3.076	...
23.6	Seventh residual	3.533	−2.419	0.354	−0.104	−2.419	−0.104	...

Note. Sampling is done "with replacement," so some residuals will appear more than once in the same sample.

TABLE 4 The first few of 255 bootstrap actuals

x Values (cost driver)	Bootstrap actual = estimate + residual	#1	#2	#3	#4	#5	#6	...
7.9	3.699 + First residual	3.595	1.594	1.365	7.233	6.775	1.281	...
8.2	4.005 + Second residual	4.359	1.586	3.901	4.359	1.670	1.586	...
9.8	5.635 + Third residual	5.530	9.168	9.168	3.530	3.300	5.989	...
11.5	7.367 + Fourth residual	4.948	10.900	4.948	4.948	5.262	4.948	...
16.4	12.358 + Fifth residual	10.253	10.253	15.892	10.253	12.712	12.712	...
19.7	15.720 + Sixth residual	13.615	19.253	13.615	13.615	13.301	18.796	...
23.6	19.693 + Seventh residual	23.226	17.274	20.047	19.588	17.274	19.588	...

Note. Each bootstrap sample is treated as if it were a set of "actual" data. The only use made of the *real* actual dataset is to calculate the estimates and residuals.

Recall that each residual is an actual cost minus its estimate. It follows that an actual equals the estimate plus the residual. Therefore, whenever a "residual" is added to an estimate, an "actual" is obtained—not a real actual, but a "bootstrap actual," namely, a number than could have been the actual if the data were collected tomorrow instead of having collected it yesterday. The first few of the 255 sets of bootstrap actuals appears in Table 4.

It is now assumed (although it is really not true) that each bootstrap actual is a real actual, and it is used to calculate a "bootstrap CER" that is not really the CER, but which could have been under other circumstances consistent with the data-gathering error-model assumptions. The first few of these bootstrap CERs are displayed in Table 5.

If all 255 bootstrap CERs are compiled and displayed together on the same graph, they form an interesting pattern; see Figure 2. That pattern is somewhat reminiscent of the region of the prediction bounds appearing in Figure 1. The pattern is narrowest in the center of the data where the point $(\bar{x}, \bar{y}) = (13.87, 9.78)$ and flares outward as the cost-driver value of interest moves to the extremes. Now an attempt must be made to estimate prediction bounds from this information. If it works for OLS, where there is the "truth" against which the bootstrap results can be tested, then there is some confidence in applying the bootstrap method to the general-error-regression case where the truth is not known.

Deriving Bounds on Estimates from Bootstrap CERs

All 255 regression lines (bootstrap CERs) in Figure 2 share in common the property that each of them could very well have been the actual CER, except for the fact that each was derived from a bootstrap sample rather than from the actual dataset. If the cost $\boldsymbol{y}$ is estimated at a cost-driver value $\boldsymbol{x} = 20$ using the actual CER $\boldsymbol{y} = \mathbf{-4.348 + 1.019}\boldsymbol{x}$, the estimate $\boldsymbol{y} = 16.032$ is obtained. This means that, if essentially random circumstances were different, there are at least 255 other numbers (bootstrap estimates) that could be the cost estimate for a cost-driver value of 20. A few of them are listed in Table 6.

TABLE 5 Bootstrap CER calculated for each bootstrap sample

x Values (cost driver)	Bootstrap actuals = estimate + residual						
	#1	#2	#3	#4	#5	#6	. . .
7.9	3.595	1.594	1.365	7.233	6.775	1.281	. . .
8.2	4.359	1.586	3.901	4.359	1.670	1.586	. . .
9.8	5.530	9.168	9.168	3.530	3.300	5.989	. . .
11.5	4.948	10.900	4.948	4.948	5.262	4.948	. . .
16.4	10.253	10.253	15.892	10.253	12.712	12.712	. . .
19.7	13.615	19.253	13.615	13.615	13.301	18.796	. . .
23.6	23.226	17.274	20.047	19.588	17.274	19.588	. . .
a (intercept)	−6.099	−3.834	−4.652	−3.366	−3.831	−7.957	. . .
b (slope)	1.114	0.998	1.045	0.897	0.897	1.242	. . .
r	0.960	0.892	0.926	0.938	0.940	0.980	. . .
r Squared	92.08%	79.55%	85.81%	87.90%	88.42%	96.07%	. . .
Standard error of estimate	2.193	3.393	2.851	2.233	2.179	1.686	. . .

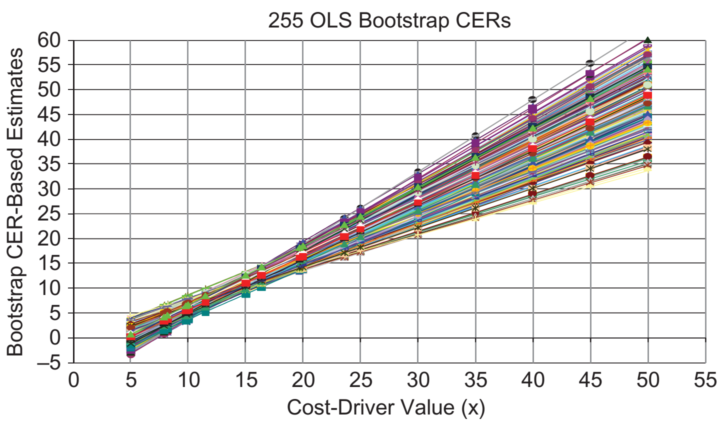

FIGURE 2 Graphs of 255 bootstrap CERs (color figure available online).

To find an 80% bootstrap interval for the cost at $x = 20$, these 255 possible estimates (or even 256 if including the actual estimate) could be used as the basis for it. As a cursory technique, the 255 bootstrap estimates could be ranked in order, smallest to largest, removing the bottom 10% and the top 10% of them and leaving the middle 80%. The shortest interval containing all those remaining numbers can be considered to be the 80% bootstrap interval for the cost. The method ultimately recommended to obtain 80% prediction bounds on the cost is based on that idea and is only slightly more complicated.

The first step in this process is to calculate the estimates given by each bootstrap CER for a range of cost-driver values. The first few of such estimates appears in Table 7.

The next step is, for each cost-driver value, to rank the 255 bootstrap estimates in order, smallest to largest. A portion of the results is displayed in Table 8. Because 255 is not an

TABLE 6 Some possible cost estimates for cost-driver value $x = 20$

Cost-driver value x	Bootstrap CER	Bootstrap estimate
20.000	$y = -6.099 + 1.114x$	16.181
20.000	$y = -3.834 + 0.998x$	16.126
20.000	$y = -4.652 + 1.045x$	16.248
20.000	$y = -3.366 + 0.897x$	14.574
20.000	$y = -3.831 + 0.897x$	14.109
20.000	$y = -7.957 + 1.242x$	16.883
20.000	•	•
20.000	•	•
20.000	•	•

TABLE 7 OLS linear bootstrap cost estimates for a range of cost-driver values

Cost-Driver (x) values	Bootstrap cost estimates (y values)						
	#1	#2	#3	#4	#5	#6	...
5	−0.526	1.154	0.574	1.119	0.655	−1.747	...
7.9	2.706	4.047	3.606	3.719	3.256	1.855	...
8.2	3.040	4.346	3.919	3.989	3.526	2.227	...
9.8	4.823	5.942	5.592	5.424	4.961	4.215	...
10	5.046	6.142	5.801	5.603	5.140	4.463	...
11.5	6.718	7.638	7.369	6.948	6.486	6.326	...
15	10.619	11.130	11.028	10.087	9.626	10.673	...
16.4	12.179	12.527	12.491	11.343	10.882	12.412	...
19.7	15.857	15.819	15.941	14.303	13.842	16.511	...
20	16.191	16.118	16.254	14.572	14.111	16.883	...
23.6	20.203	19.710	20.017	17.801	17.341	21.355	...
25	21.764	21.106	21.481	19.056	18.597	23.094	...
30	27.336	26.095	26.707	23.541	23.083	29.304	...
35	32.908	31.083	31.934	28.025	27.568	35.514	...
40	38.481	36.071	37.161	32.509	32.054	41.724	...
45	44.053	41.059	42.387	36.994	36.539	47.934	...
50	49.626	46.047	47.614	41.478	41.025	54.144	...

exact multiple of 10, the lower 10th percentile value must be interpolated between the 25th and 26th ranks and the upper 10th percentile value between the 229th and 230th ranks.

Table 9 shows a selection portion of Table 8, along with the interpolated lower and upper 10% bootstrap bounds (BBs) on cost estimates at cost-driver values $\boldsymbol{x} = 5$, 15, and 50. The 80% bootstrap interval runs from the lower 10% bound to the upper 10% bound; 80% of the bootstrap estimates lie between those two numbers.

Do OLS BBs Match OLS Prediction Bounds?

Upon closer examination, it turns out that the BBs, namely the 10th and 90th percentile bootstrap estimates, for $\boldsymbol{x}$ values within the range of the database are closer together than

TABLE 8 A portion of the ranked bootstrap estimates for cost-driver values $x = 5$, 15, and 50

	Estimate ranks	$x = 5$	$x = 15$	$x = 50$	Estimate ranks	$x = 5$	$x = 15$	$x = 50$	
	1	−3.405	8.709	33.719	226	2.591	12.013	54.701	
	2	−3.129	9.205	34.271	227	2.634	12.042	54.723	
	3	−3.066	9.251	34.774	228	2.683	12.051	54.728	
	4	−3.030	9.325	35.090	229	2.697	12.068	55.050	← Upper
	5	−2.971	9.330	35.278	230	2.725	12.077	55.120	10th
	6	−2.784	9.384	35.572	231	2.762	12.085	55.324	Percentile
	7	−2.684	9.528	36.138	232	2.772	12.102	55.418	
	8	−2.683	9.559	36.257	233	2.783	12.114	55.441	
	9	−2.670	9.594	36.425	234	2.820	12.133	55.784	
	10	−2.516	9.626	36.523	235	2.825	12.136	55.944	
	11	−2.419	9.655	38.009	236	2.829	12.149	56.315	
	12	−2.399	9.662	38.030	237	2.866	12.187	56.329	
	13	−2.108	9.693	38.144	238	2.979	12.209	56.353	
	14	−2.087	9.718	38.320	239	3.016	12.212	56.409	
	15	−1.949	9.728	38.522	240	3.053	12.275	56.675	
	16	−1.924	9.745	38.663	241	3.098	12.282	56.755	
	17	−1.860	9.751	38.784	242	3.170	12.283	57.040	
	18	−1.798	9.754	39.420	243	3.172	12.330	57.150	
	19	−1.764	9.762	39.637	244	3.208	12.439	57.655	
	20	−1.747	9.766	39.679	245	3.338	12.474	58.021	
	21	−1.715	9.781	39.717	246	3.379	12.513	58.310	
	22	−1.554	9.784	39.735	247	3.380	12.563	58.500	
	23	−1.551	9.792	39.826	248	3.591	12.591	58.892	
	24	−1.521	9.795	39.869	249	3.603	12.756	59.999	
	25	−1.517	9.850	39.883	250	3.801	12.769	60.250	
Lower →	26	−1.499	9.855	39.887	251	3.869	12.807	60.259	
10th	27	−1.498	9.888	39.899	252	3.872	12.946	60.381	
Percentile	28	−1.461	9.889	40.219	253	3.878	13.060	60.422	
	29	−1.452	9.893	40.264	254	3.925	13.278	62.543	
	30	−1.371	9.895	40.513	255	4.511	13.281	62.568	

TABLE 9 Interpolated lower and upper 10% BBs

Estimate ranks	$x = 5$	$x = 15$	$x = 50$
•	•	•	•
Rank 25	−1.517	9.850	39.883
Lower 10% bound	−1.508	9.852	39.885
Rank 26	−1.499	9.855	39.887
•	•	•	•
Rank 229	2.697	12.068	55.050
Upper 10% bound	2.711	12.072	55.085
Rank 230	2.725	12.077	55.120
•	•	•	•

TABLE 10 Lower 10% B-BBs versus OLS prediction bounds

Cost-driver (x) Values	Bootstrap 80% lower bounds	B-B 80% lower bounds (B-BLBs)	OLS prediction bounds (80% lower bounds)	Differences: B-BB versus OLS	
				Absolute	Percentage
5	−1.508	−4.258	−4.213	0.0447	1.0604%
7.9	1.989	−0.761	−0.931	0.1700	18.2625%
8.2	2.330	−0.420	−0.597	0.1773	29.6875%
9.8	4.174	1.424	1.158	0.2659	22.9604%
10	4.426	1.676	1.375	0.3008	21.8762%
11.5	6.146	3.396	2.980	0.4159	13.9546%
15	9.852	7.102	6.582	0.5200	7.9003%
16.4	11.193	8.443	7.966	0.4774	5.9934%
19.7	14.069	11.319	11.103	0.2163	1.9476%
20	14.338	11.588	11.380	0.2073	1.8211%
23.6	17.516	14.766	14.618	0.1479	1.0117%
25	18.838	16.087	15.838	0.2499	1.5781%
30	23.143	20.392	20.058	0.3340	1.6651%
35	27.434	24.684	24.129	0.5550	2.3002%
40	31.604	28.854	28.104	0.7496	2.6674%
45	35.783	33.033	32.018	1.0151	3.1704%
50	39.885	37.135	35.891	1.2441	3.4662%

the known 80% lower and upper OLS prediction bounds. On the other hand, far outside the range of the cost-driver data, the BBs are farther apart—this is probably due to lack of a normal-distribution assumption on the estimating error. Some theoretical analysis and numerical experimentation indicates that adjusting the lower BBs downward and the upper BBs upward by an additive amount equal to the SEE of the "real" CER leads to "bootstrap-based" bounds (B-BBs) that are significantly closer to the known OLS prediction bounds. This issue is discussed in detail in the next few paragraphs.

In Tables 10 and 11, the term "absolute" difference refers to the absolute value of the dollar-valued difference between the OLS bounds and the B-BBs. The percentage difference is the absolute difference expressed as a percentage of the OLS value.

Comparison of B-BB bounds with OLS prediction bounds in Tables 10 and 11 illustrates, for the example being worked with, that the adjusted

$$\boldsymbol{B\text{-}BB(y) = BB(y) \pm StdError} \tag{2}$$

bounds seem pretty good (not perfect, but better than without the adjustment and certainly better than nothing) in the OLS case. Prediction-bound formulas do not exist in any non-OLS cost-modeling context, even for linear CERs; thus, that adjustment will be used here for all non-OLS additive-error-CER scenarios. As an approximating technique, it will not yield "exact" prediction bounds (those are yet unknown), but it appears to provide adequate results.

For the given example, Figure 3 displays graphically the comparison between the B-BBs and the OLS prediction bounds to which they correspond. Of course, it is not recommended to use the B-BBs in the OLS case, because exact closed-form expressions for the bounds are available for OLS. This study should be considered a demonstration in

TABLE 11 Upper 10% B-BBs versus OLS prediction bounds

Cost-driver (x) values	Bootstrap 80% upper bounds (BUBs)	B-B 80% upper bounds (B+BUBs)	OLS prediction bounds (80% upper bounds)	Differences: B-BB vs. OLS	
				Absolute	Percentage
5	2.711	5.461	5.704	0.2428	4.2569%
7.9	5.215	7.965	8.329	0.3643	4.3738%
8.2	5.452	8.202	8.607	0.4051	4.7060%
9.8	6.906	9.656	10.112	0.4557	4.5070%
10	7.096	9.846	10.302	0.4559	4.4254%
11.5	8.501	11.251	11.753	0.5016	4.2675%
15	12.072	14.823	15.282	0.4593	3.0055%
16.4	13.593	16.343	16.751	0.4081	2.4363%
19.7	17.621	20.371	20.337	0.0346	0.1700%
20	17.996	20.746	20.671	0.0752	0.3636%
23.6	22.250	25.000	24.768	0.2321	0.9371%
25	23.990	26.740	26.400	0.3395	1.2859%
30	30.217	32.967	32.366	0.6011	1.8570%
35	36.465	39.215	38.483	0.7325	1.9033%
40	42.657	45.407	44.694	0.7134	1.5963%
45	48.922	51.672	50.967	0.7055	1.3842%
50	55.085	57.835	57.281	0.5541	0.9674%

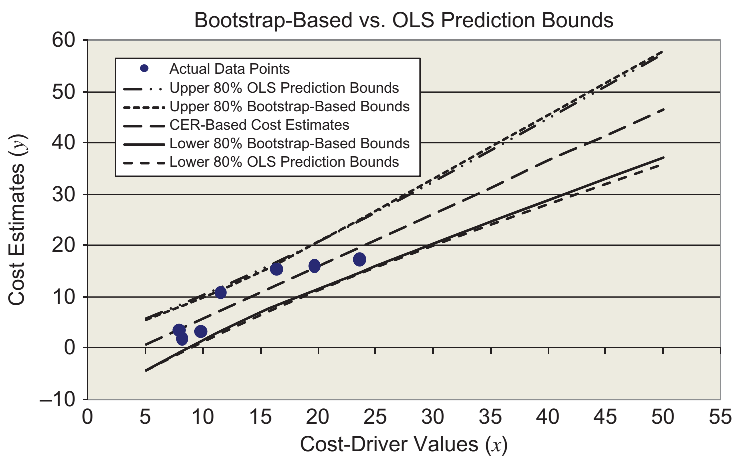

FIGURE 3 Bootstrap-based bounds graphed together with their corresponding OLS prediction bounds (color figure available online).

principle; because this seems to work for OLS, it will be applied to non-OLS situations, such as those involving non-linear CER forms and multiplicative-error models.

Extension of Bootstrap Method to Multiplicative-Error CERs

Figure 4 illustrates the difference in behavior of standard errors between additive-error and multiplicative-error CERs. Additive-error CERs have constant standard error across

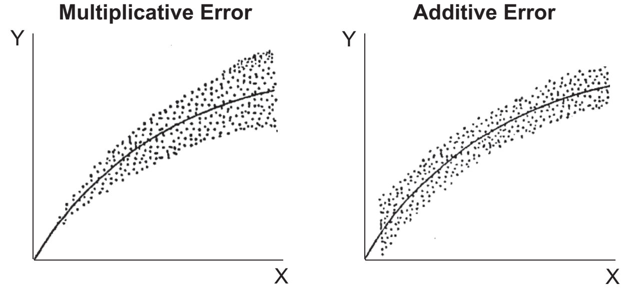

FIGURE 4 Estimates ± one standard error for multiplicative-error and additive-error CERs, respectively (Eskew & Lawler, 1994).

the range of cost-driver values. This kind of error model is adequate if the cost estimates corresponding to the cost-driver values do not range over an order of magnitude or more. However, suppose that the cost estimates based on a CER vary over a large range, say from \$100K to \$1,000K. Then a standard error of \$50,000 would equate to a very unacceptable 80% error of estimation at the low end, but a fabulous 8% error at the high end. The obvious solution to the problem of a CER that generates wide-ranging estimates is to use a multiplicative-error model that expresses the error as a percentage of the estimate. In most situations, percentage error is what a decision maker wants to understand anyway.

Applying B-BBs in the case of multiplicative-error CERs differs slightly from how they are applied in the additive-error situation. In the additive-error situation, they are defined as in Equation (2), namely, $\boldsymbol{B\text{-}BB = BB \pm StdError}$. In the multiplicative-error situation, the standard error is expressed as a percentage of the estimate, and this leads us to define B-BBs as follows:

$$\boldsymbol{B\text{-}BB(y) = BB(y) \pm (\%StdError) \times ESTy.} \tag{3}$$

A Brief Introduction to General-Error Regression

In theory, there is no limit to the number of algebraic forms that may serve as models for CERs. In practice, though, in the one-cost-driver situation, most CERs take one of very few algebraic forms. Primary among these are the following:

- factor CERs of the form $\boldsymbol{y = ax}$,
- linear CERs of the form $\boldsymbol{y = a + bx}$, and
- non-linear CERs of the forms
  - $\boldsymbol{y = ax^b}$,
  - $\boldsymbol{y = ab^x}$,
  - $\boldsymbol{y = a + bx^c}$,

where $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are constant coefficients or exponents derived from the historical data supporting the CER. Although the case of only one cost driver per CER is discussed here, the concepts are the same for multiple cost drivers, but the mathematical details are more complicated.

OLS offers the opportunity to derive only two kinds of CERs in the one-cost-driver situation: (1) linear additive-error CERs that are unbiased in the statistical sense and (2) log-log multiplicative-error CERs that are not unbiased in either the additive-error or multiplicative-error sense. In addition, if the analyst chooses the OLS linear form, he/she ipso facto must use an additive-error model. On the other hand, if he/she were to choose the OLS log-log form, the error model must be multiplicative.

A major statistical breakthrough occurred in 1974, when Wedderburn (1974) enhanced an existing 1972 technique, of which he was one of the inventors, that came to be known as iteratively reweighted least squares (IRLS). IRLS is a technique for deriving CERs with the following characteristics:

- are "almost" unbiased in the analyst's choice of either the additive-error or multiplicative-error sense; for a discussion of the word "almost" in this context, see Goldberg and Tuow (2003, p. 65) or Sperling and Goldberg (n.d., p. 7);
- allow the analyst his/her choice of the additive-error or multiplicative-error model independent of the algebraic form of the CER; and
- allow the analyst his/her choice of algebraic functional form $y = f(a, b, c, d)$ independent of the CERs error model.

In a more recent refinement of IRLS, Book and Lao (1998) introduced the minimum percentage error–zero percentage bias (MPE-ZPB or ZMPE) technique to derive CERs that also had minimum possible percentage (i.e., multiplicative) error among all unbiased CERs of the algebraic form being considered.

For both unbiasedness and minimum possible percentage standard error in CERs, ZMPE offers a slight improvement over IRLS (Book, 2006b). Experience and theory indicate that IRLS CERs do not necessarily have the minimum possible standard error among all zero-bias CERs. In fact, Wedderburn claimed neither unbiasedness nor minimum percentage error for IRLS CERs. He credited IRLS CERs with having a desirable statistical property called "maximum quasi-likelihood." To avoid statistical detail, simply note that OLS has a related property called "maximum likelihood" in addition to its unbiasedness and minimum standard error. In the sections that follow, the ZMPE method will be applied to derive the multiplicative-error CERs and will use the $(\boldsymbol{x}, \boldsymbol{y})$ dataset of Table 2 for all (illustrative only) examples.

Prediction Bounds for a Multiplicative-Error Factor CER

Multiplicative-error factor CERs have the algebraic form $\boldsymbol{y} = \boldsymbol{ax\varepsilon}$, where $\boldsymbol{y}$ is cost, $\boldsymbol{a}$ is the CER's single coefficient derived from the supporting historical database, $\boldsymbol{x}$ is the value of the cost driver (e.g., weight), and $\boldsymbol{\varepsilon}$ is the multiplicative error, which is expressed as a percentage of the estimate $\boldsymbol{ax}$. ZMPE CERs minimize the percentage standard error subject to the constraint that percentage bias be zero. Specifically, the sum of percentage squared errors

$$\boldsymbol{F(a)} = \sum_{k=1}^{n} \left(\frac{\boldsymbol{y_k - ax_k}}{\boldsymbol{ax_k}} \right)^{\mathbf{2}} \tag{4}$$

is minimized, subject to the constraint that percentage bias

$$\boldsymbol{B(a)} = \sum_{k=1}^{n} \left(\frac{\boldsymbol{ax_k - y_k}}{\boldsymbol{ax_k}} \right) = \mathbf{0}. \tag{5}$$

In most ZMPE and IRLS cases, Excel's Solver routine or some more advanced numerical optimization procedure is used to find the numerical value of the coefficient $\boldsymbol{a}$. But, as will be seen, this is not necessary in the simple case of a one-cost-driver factor CER.

The equation $\boldsymbol{B(a) = 0}$ is first rearranged as follows:

$$\begin{aligned} \boldsymbol{0 = B(a)} &= \sum_{k=1}^{n}\left(\frac{ax_k - y_k}{ax_k}\right) = \sum_{k=1}^{n}\left(1 - \frac{y_k}{ax_k}\right) \\ &= \sum_{k=1}^{n} 1 - \sum_{k=1}^{n}\frac{y_k}{ax_k} = n - \frac{1}{a}\sum_{k=1}^{n}\frac{y_k}{x_k}. \end{aligned} \tag{6}$$

It then follows that the algebraic form of the the ZMPE factor CER's zero-bias constraint is

$$\sum_{k=1}^{n}\frac{y_k}{x_k} = na, \tag{7}$$

from which it follows uniquely that the ZMPE numerical value of the coefficient $\boldsymbol{a}$ is

$$a = \frac{1}{n}\sum_{k=1}^{n}\frac{y_k}{x_k}. \tag{8}$$

Table 12 displays the calculations required when applying Equation (8) instead of the Excel Solver to derive the ZMPE factor CER for the dataset of Table 2.

Table 12 results in the calculation that the ZMPE factor is $\boldsymbol{a = 0.637}$, so the factor CER is $\boldsymbol{y = 0.637x}$, which is graphed along the supporting data points in Figure 5.

The next phase is a bootstrap sampling process that is headed toward estimating prediction bounds for the ZMPE factor CER-based estimates. Table 13 contains the initial step in this direction, the computations of the multiplicative-error residuals. The key idea in that step is to understand that residuals of the multiplicative-error CER are ratios, rather than differences, of the estimates and their corresponding actuals. That is, for the CER $\boldsymbol{y = ax}$, the estimate $\boldsymbol{ESTy_i = ax_i}$ leads to the residual $\boldsymbol{y_i/ESTy_i = y_i/ax_i}$.

Using the residuals in Table 13, 255 sets of bootstrap residuals are generated. For each such set, a set of bootstrap actuals are calculated and then ZMPE bootstrap factor CERs are derived from each set of bootstrap actuals. The results of this process are displayed in abbreviated form in Table 14.

TABLE 12 Derivation of ZMPE factor CER using Equation (8)

Number of data points	Diameter x (feet)	Cost y (FY \$99K)	y/x	a	$ESTy = ax$	% Bias	Squared % error
7	7.9	3.595	0.455	0.637	5.034	28.5826%	8.1696%
	8.2	1.900	0.232	0.637	5.225	63.6360%	40.4954%
	9.8	3.300	0.337	0.637	6.244	47.1530%	22.2341%
	11.5	10.900	0.948	0.637	7.328	−48.7514%	23.7670%
	16.4	15.434	0.941	0.637	10.450	−47.6954%	22.7485%
	19.7	16.074	0.816	0.637	12.553	−28.0531%	7.8698%
	23.6	17.274	0.732	0.637	15.038	−14.8717%	2.2117%
Sums	97.1	68.477	4.460		61.871	0.0000%	127.4960%

% bias = 0.0000%
%std error = 46.0970%
R^2 = 86.0636%

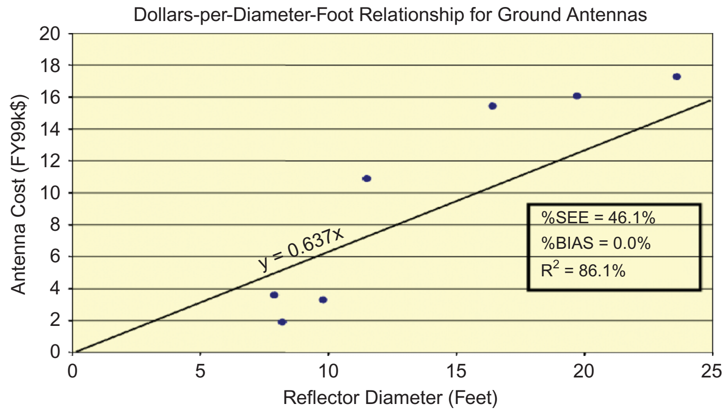

FIGURE 5 ZMPE factor CER versus data points (color figure available online).

TABLE 13 Calculation of ZMPE factor CER residuals

x Values	Actual y values	Estimated y values	Residuals = actual/estimated
7.9	3.595	5.034	0.714
8.2	1.900	5.225	0.364
9.8	3.300	6.244	0.528
11.5	10.900	7.328	1.488
16.4	15.434	10.450	1.477
19.7	16.074	12.553	1.281
23.6	17.274	15.038	1.149

TABLE 14 The path from ZMPE factor CER residuals to bootstrap ZMPE factor CERs

x Values (cost driver)	Estimated y values	Bootstrap y value = (estimated y value) × (bootstrap residual)						
		1	2	3	4	5	6	. . .
7.9	5.034	1.830	5.782	5.782	2.660	6.446	1.830	. . .
8.2	5.225	2.761	7.772	1.900	6.691	7.717	7.717	. . .
9.8	6.244	3.300	7.996	7.173	4.460	9.289	9.289	. . .
11.5	7.328	8.417	5.233	9.383	10.900	2.665	9.383	. . .
16.4	10.450	15.544	3.800	7.463	13.381	15.434	5.522	. . .
19.7	12.553	14.419	8.965	16.074	16.074	18.540	14.419	. . .
23.6	15.038	22.210	7.947	22.369	10.739	5.468	7.947	. . .
a (factor)	0.637	0.608	0.568	0.676	0.663	0.721	0.620	. . .
% Standard Error	46.10%	49.59%	46.86%	36.52%	36.12%	46.82%	49.81%	. . .
% Bias	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	. . .
R^2	86.06%	94.92%	5.18%	84.04%	56.53%	12.61%	18.29%	. . .

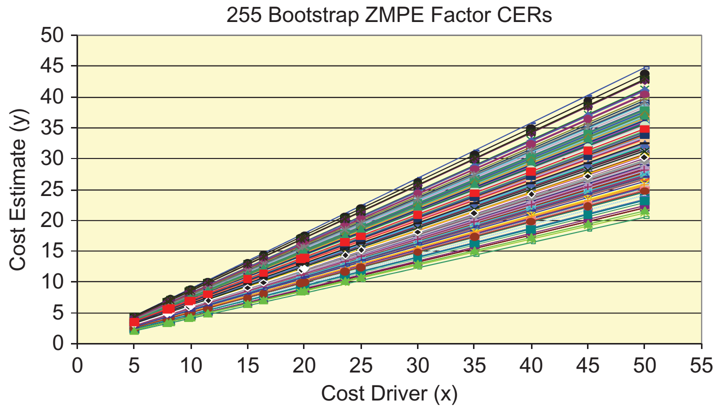

FIGURE 6 Two hundred fifty-five bootstrap ZMPE factor CERs (color figure available online).

TABLE 15 Points on the bootstrap ZMPE factor CERs for a range of cost-driver values

Cost-driver (x) values	Bootstrap cost estimates (y values)						
	#1	#2	#3	#4	#5	#6	. . .
20	49.659	49.988	60.788	55.820	38.285	39.057	. . .
30	69.243	69.354	77.751	74.620	60.518	60.706	. . .
40	88.826	88.720	94.714	93.421	82.751	82.354	. . .
50	108.410	108.086	111.677	112.221	104.984	104.003	. . .
53	114.285	113.895	116.766	117.861	111.654	110.497	. . .
55	118.202	117.769	120.158	121.621	116.101	114.827	. . .
60	127.994	127.452	128.640	131.022	127.217	125.651	. . .
61	129.952	129.388	130.336	132.902	129.441	127.816	. . .
64	135.828	135.198	135.425	138.542	136.111	134.310	. . .
66	139.744	139.071	138.818	142.302	140.557	138.640	. . .
67.3	142.290	141.589	141.023	144.746	143.448	141.454	. . .
70	147.578	146.817	145.603	149.822	149.451	147.300	. . .
71	149.536	148.754	147.299	151.702	151.674	149.464	. . .
76	159.328	158.437	155.780	161.102	162.791	160.289	. . .
78	163.245	162.310	159.173	164.863	167.237	164.618	. . .
79	165.203	164.247	160.869	166.743	169.461	166.783	. . .
80	167.162	166.183	162.566	168.623	171.684	168.948	. . .
90	186.746	185.549	179.529	187.423	193.917	190.596	. . .
100	206.329	204.915	196.492	206.224	216.150	212.245	. . .
110	225.913	224.281	213.454	225.024	238.383	233.893	. . .
120	245.497	243.646	230.417	243.825	260.617	255.542	. . .

The 255 bootstrap ZMPE CERs are graphed together with the supporting data points in Figure 6, the outlines of which illustrate the trend of the prediction bounds. Notice the obviously widening bounds in the direction of increasing estimates. This pattern is a consequence of multiplicative errors being a percentage of the estimate.

The first few actual bootstrap cost estimates on which Figure 6 is based are displayed in Table 15.

TABLE 16 Portion of the ranked bootstrap estimates for cost-driver values $x = 5$, 15, and 50

Estimate ranks	$x = 5$	$x = 15$	$x = 50$	Estimate ranks	$x = 5$	$x = 15$	$x = 50$
1	2.045	6.135	20.451	226	3.782	11.346	37.820
2	2.120	6.360	21.200	227	3.786	11.358	37.860
3	2.153	6.458	21.527	228	3.787	11.361	37.869
4	2.195	6.585	21.950	229	3.792	11.375	37.917
5	2.205	6.614	22.045	230	3.796	11.388	37.961
6	2.209	6.628	22.094	231	3.801	11.403	38.010
7	2.229	6.687	22.290	232	3.842	11.526	38.420
8	2.280	6.839	22.795	233	3.875	11.626	38.754
9	2.289	6.867	22.890	234	3.880	11.641	38.802
10	2.308	6.923	23.076	235	3.881	11.642	38.807
11	2.317	6.951	23.171	236	3.890	11.671	38.904
12	2.322	6.966	23.219	237	3.900	11.700	39.000
13	2.331	6.994	23.315	238	3.910	11.729	39.096
14	2.337	7.011	23.369	239	3.914	11.743	39.144
15	2.337	7.011	23.369	240	3.941	11.822	39.407
16	2.341	7.023	23.411	241	3.946	11.837	39.455
17	2.402	7.207	24.023	242	3.980	11.939	39.798
18	2.407	7.221	24.072	243	4.030	12.090	40.301
19	2.412	7.236	24.120	244	4.045	12.134	40.446
20	2.453	7.358	24.528	245	4.105	12.314	41.046
21	2.460	7.379	24.597	246	4.124	12.371	41.238
22	2.477	7.430	24.767	247	4.134	12.402	41.339
23	2.482	7.447	24.822	248	4.267	12.801	42.669
24	2.486	7.459	24.863	249	4.283	12.850	42.833
25	2.491	7.473	24.911	250	4.283	12.850	42.833
26	2.491	7.473	24.911	251	4.288	12.864	42.879
27	2.496	7.489	24.965	252	4.288	12.864	42.881
28	2.520	7.559	25.197	253	4.293	12.878	42.927
29	2.520	7.559	25.197	254	4.378	13.133	43.775
30	2.535	7.604	25.347	255	4.472	13.415	44.715

Table 16 exhibits a portion of the ranked bootstrap estimates for the cost-driver values $\boldsymbol{x} = 5$, 15, and 50. It is from Table 16 that the lower 10% prediction bounds are obtained by interpolating between the 25th and 26th lowest estimates, and the upper 10% prediction bounds are obtained between the 229th and 230th lowest estimates, for those cost-driver values. The results of the interpolation appear in Table 17.

These interpolated lower and upper 10% BBs serve as the basis of approximate 80% bootstrap-based (B-B) prediction bounds derived by applying Equation (3). For this example, those bounds are displayed in Table 18. The resulting prediction bounds are illustrated on a Cartesian graph, together with the CER and the supporting data points, in Figure 7.

Prediction Bounds for a Multiplicative-Error Linear CER

Now that a relatively thorough treatment of the multiplicative-error factor-CER case has been offered, what happens if the same technique for multiplicative-error CERs is

TABLE 17 Interpolated lower and upper 10% BBs

Estimate ranks	$x = 5$	$x = 15$	$x = 50$
•	•	•	•
Rank 25	2.491	7.473	24.911
Lower 10% bounds	2.491	7.473	24.911
Rank 26	2.491	7.473	24.911
•	•	•	•
Rank 229	3.792	11.375	37.917
Upper 10% bounds	3.794	11.382	37.939
Rank 230	3.796	11.388	37.961
•	•	•	•

TABLE 18 Calculation of B-B prediction bounds for multiplicative-error CERs

Cost driver x-values	Lower 80% BBs	Lower *BB* - % *SEE* × *ESTy*	ZMPE factor CER estimate	Upper *BB* + %*SEE* × *ESTy*	Upper 80% BBs
5	2.491	1.023	3.186	5.263	3.794
7.9	3.936	1.616	5.034	8.315	5.994
8.2	4.085	1.677	5.225	8.631	6.222
9.8	4.883	2.004	6.244	10.315	7.436
10	4.982	2.045	6.372	10.525	7.588
11.5	5.730	2.352	7.328	12.104	8.726
15	7.473	3.068	9.558	15.788	11.382
16.4	8.171	3.354	10.450	17.261	12.444
19.7	9.815	4.029	12.553	20.734	14.948
20	9.965	4.090	12.744	21.050	15.176
23.6	11.758	4.826	15.038	24.839	17.907
25	12.456	5.113	15.930	26.313	18.970
30	14.947	6.135	19.116	31.575	22.763
35	17.438	7.158	22.302	36.838	26.557
40	19.929	8.180	25.488	42.100	30.351
45	22.420	9.203	28.673	47.363	34.145
50	24.911	10.225	31.859	52.625	37.939

applied for other algebraic forms will be described, beginning with the multiplicative-error linear CER.

Table 19 displays the calculations needed to run Excel's Solver routine to derive a multiplicative-error CER that offers minimum percentage error and zero percentage bias. The resulting CER is graphed along with the data points from which it was derived in Figure 8.

The next step in the process is to calculate the residuals and set them up for bootstrap sampling. Table 20 compares the actual costs with their corresponding estimates according to the ZMPE linear CER and then calculates the CER's residuals. Table 21 displays the first six of the 255 sets of bootstrap residuals, from which the 255 bootstrap ZMPE linear CERs are derived. The 255 linear bootstrap CERs themselves are graphed in Figure 9 and illustrate the flaring-out pattern that is characteristic of multiplicative-error regression.

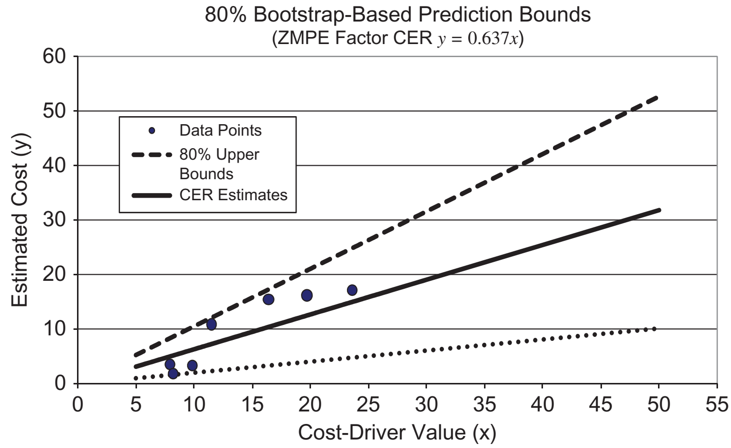

FIGURE 7 Approximate 80% bootstrap-based prediction bounds for ZMPE CER, by analogy with OLS (color figure available online).

TABLE 19 Excel solver setup and derivation of ZMPE linear CER

x values	Actual y values	Estimated y values	Percentage bias	Percentage squared error	Actual y^2	Estimated y^2	Actual y *estimated y
7.9	3.595	2.852	−26.0399%	6.7808%	12.924	8.135	10.254
8.2	1.9	3.206	40.7414%	16.5986%	3.610	10.280	6.092
9.8	3.3	5.094	35.2226%	12.4063%	10.890	25.953	16.811
11.5	10.9	7.100	−53.5112%	28.6345%	118.810	50.417	77.395
16.4	15.434	12.883	−19.8040%	3.9220%	238.208	165.964	198.832
19.7	16.074	16.777	4.1896%	0.1755%	258.373	281.464	269.672
23.6	17.274	21.379	19.2014%	3.6869%	298.391	457.065	369.302
97.1	68.477	69.292	0.0000%	72.2047%	941.207	999.278	948.358

$a = -6.470138$	$n = 7$
$b = 1.180052$	$n\sum x^2 - (\sum x)^2 = 1899.3490$
%std error = 38.0012%	$n\sum y^2 - (\sum y)^2 = 2193.5568$
%bias = 0.0000%	$n\sum xy - (\sum x)^2(\sum y)^2 = 1893.5931$
$R^2 = 86.0636\%$	$\therefore R = 0.927705$

After the bootstrap estimates are calculated for a range of cost-driver values and ranked from smallest to largest, the 10th and 90th percentile values, respectively, are marked, adjusted as described earlier, and then each sequence is connected by a curve that passes though the percentile values to form, respectively, the lower and upper approximate 80% prediction bounds. The approximate prediction bounds, which parallel the flaring out to the right visible in Figure 9, are displayed in Figure 10.

Prediction Bounds for a Multiplicative-Error Power CER

A power CER for cost $\boldsymbol{y}$ in terms of a cost-driver value $\boldsymbol{x}$ has the algebraic form $\boldsymbol{y = ax^b}$. CERs of this form have historically been derived using log-log OLS, i.e., by taking logarithms of both sides of the equation and solving the resulting linear regression problem by

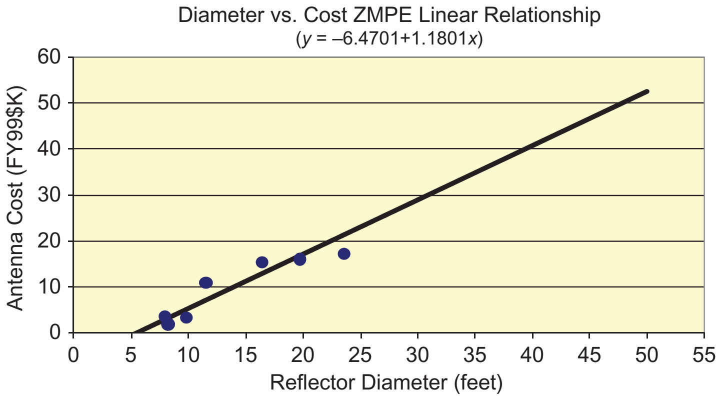

FIGURE 8 ZMPE linear CER graphed together with its supporting database (color figure available online).

TABLE 20 Multiplicative-error residuals of the ZMPE linear CER

x values (cost drivers)	y values (actual costs)	Estimated y values (estimated costs)	Residuals = actual/estimated
7.9	3.595	2.852	1.260
8.2	1.900	3.206	0.593
9.8	3.300	5.094	0.648
11.5	10.900	7.100	1.535
16.4	15.434	12.883	1.198
19.7	16.074	16.777	0.958
23.6	17.274	21.379	0.808
$a = -6.4701$		$b = 1.1801$	
% SEE = 38.0012%		% bias = 0.0000%	

TABLE 21 The first six of the 255 sets of bootstrap residuals

x Values	Samples	#1	#2	#3	#4	#5	#6	. . .
7.9	First residual	0.648	0.808	0.808	1.260	0.958	0.648	. . .
8.2	Second residual	1.260	1.535	0.648	0.958	1.198	1.198	. . .
9.8	Third residual	1.260	0.958	0.808	0.593	1.535	1.535	. . .
11.5	Fourth residual	0.808	0.593	0.958	1.535	0.648	0.958	. . .
16.4	Fifth residual	1.535	0.648	0.593	0.958	1.198	1.260	. . .
19.7	Sixth residual	0.808	0.593	0.958	0.958	1.198	0.808	. . .
23.6	Seventh residual	1.198	1.260	1.535	0.593	0.648	1.260	. . .

OLS methods. However, this 18th century log-log technique has a number of weaknesses with respect to current statistical capabilities. Those weaknesses are summarized in some detail in Book (2010). The ZMPE power CER will be used for the database of this study; both are graphed in Figure 11.

Table 22, analogous to several previous tables, compares the actual costs with the ZMPE power CER-based estimates and calculates the multiplicative-error residuals.

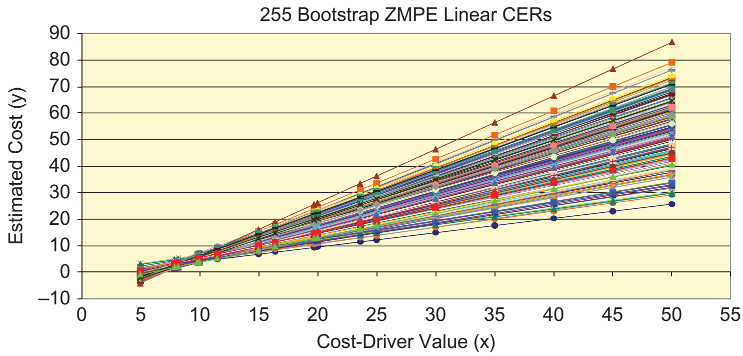

FIGURE 9 Bootstrap CERs derived from residuals of the ZMPE linear CER (color figure available online).

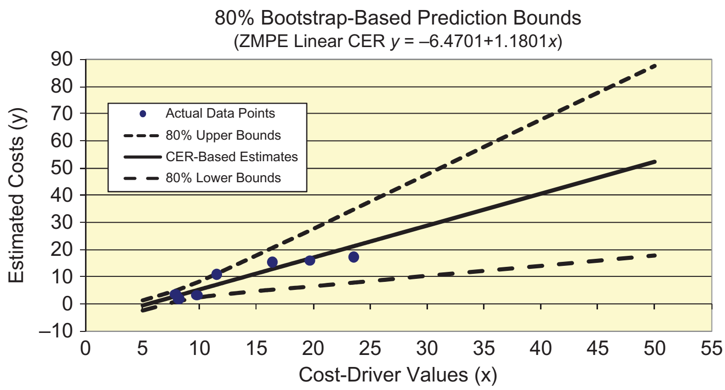

FIGURE 10 Approximate 80% upper and lower prediction bounds on estimates based on the ZMPE linear CER (color figure available online).

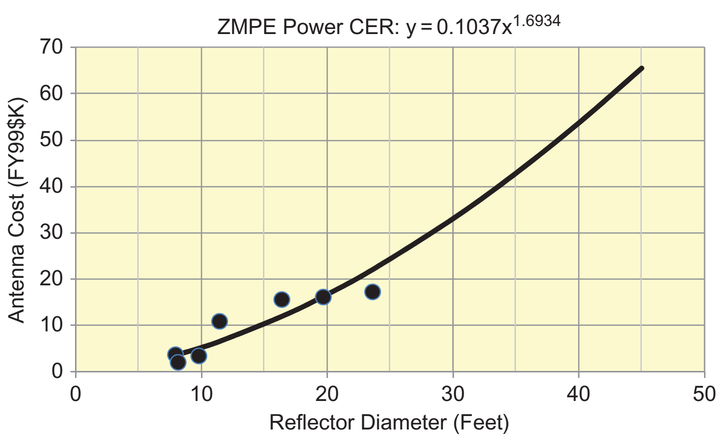

FIGURE 11 ZMPE power CER together with its supporting database (color figure available online).

TABLE 22 Multiplicative-error residuals of the ZMPE power CER

x Values (cost drivers)	y Values (actual costs)	Estimated y Values (cost estimates)	Residuals = actual/estimate
7.9	3.595	3.436	1.046
8.2	1.900	3.659	0.519
9.8	3.300	4.949	0.667
11.5	10.900	6.489	1.680
16.4	15.434	11.835	1.304
19.7	16.074	16.144	0.996
23.6	17.274	21.921	0.788
$a = 0.103734$		$b = 1.693439$	
% SEE = 43.450063%		% bias = 0.000000%	

The entire bootstrap process will not be reviewed again that consists of generating (1) 255 sets of bootstrap residuals, (2) 255 sets of bootstrap actuals, (3) 255 bootstrap ZMPE power CERs, (4) ranking of 255 bootstrap estimates for a range of cost-driver values, (5) identification of the 10th and 90th percentile estimates for each such cost-driver value, and (6) adjusting those estimates and connecting them as before to form approximate 80% lower and upper prediction bounds on power CER-based estimates. The 255 bootstrap power CERs are displayed in the graph of Figure 12.

The final result of that process is the set of approximate 80% prediction bounds that are illustrated in Figure 13, along with the supporting data points and the ZMPE power CER. The widening of the prediction interval in this case is, as before, the most spectacular effect of choosing the multiplicative-error CER model.

Prediction Bounds for a Multiplicative-Error Triad CER

One apparent fact from Figure 11 is that the concavity of the power CER is opposite from that of its supporting database. Notice that the CER slopes upward to the right, while the

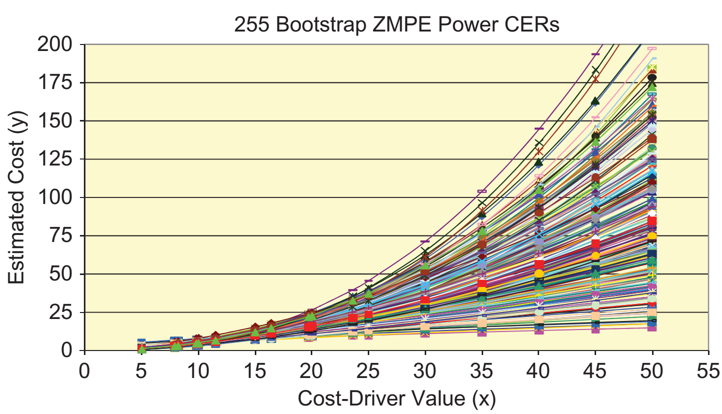

FIGURE 12 Two hundred fifty-five ZMPE bootstrap power CERs (color figure available online).

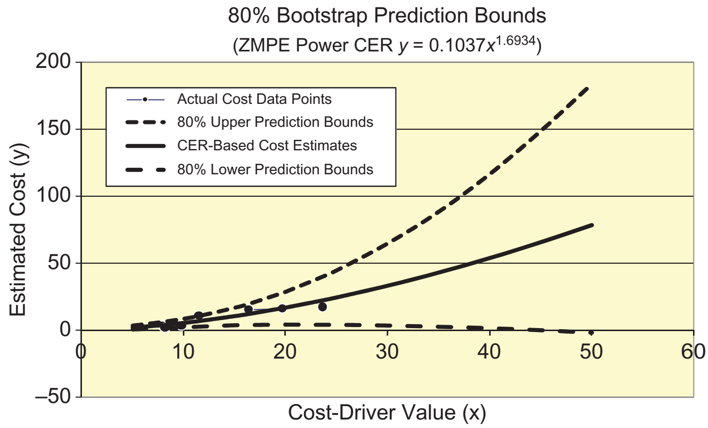

FIGURE 13 Approximate 80% upper and lower prediction bounds on estimates based on the ZMPE power CER (color figure available online).

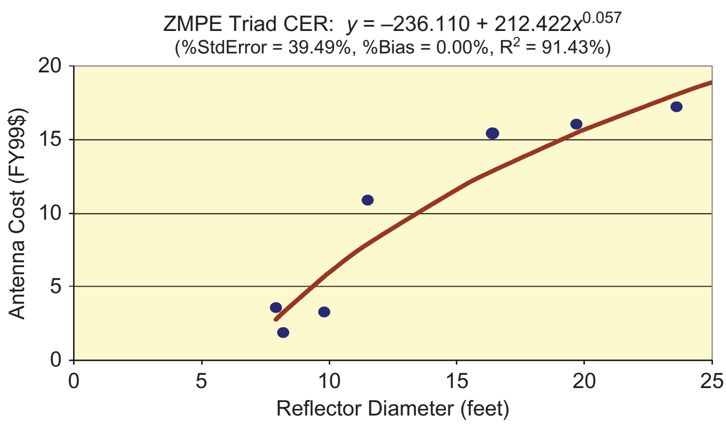

FIGURE 14 ZMPE triad CER together with its supporting database (color figure available online).

TABLE 23 Multiplicative-error residuals of the ZMPE triad CER

x Values	Actual y values	Estimated y values	Residuals = actual/estimated
7.9	3.595	2.786965	1.2899336721
8.2	1.900	3.293516	0.5768911567
9.8	3.300	5.730955	0.5758203023
11.5	10.900	7.939498	1.3728828226
16.4	15.434	12.912169	1.1953065162
19.7	16.074	15.520302	1.0356757065
23.6	17.274	18.116607	0.9534898236
97.1	68.477	66.3000109	7.000000000
$a = -236.110$		$b = 212.422$	
% SEE = 39.4852%		% bias = 0.00000000000%	

sequence of data points slopes downward to the right. This feature seriously reduces the credibility of the power CER in this case, and there is a risk of its occurring anytime a power CER is derived. The reason for this is that a power CER must pass through the (0,0) (***x,y***) point on the graph of Figure 11, even though the natural ***y***-axis point to pass through may be elsewhere. This problem is completely solved by use of the triad CER form $\boldsymbol{y = a + bx^c}$, where coefficient ***a*** (the fixed cost) identifies the point at which the CER intersects the ***y***-axis.

The power CER has no fixed-cost term; therefore, the fixed cost is effectively considered to be zero, and so the CER must pass through the (0,0) points.

The nearly vertical bootstrap CER on the left side of Figure 15 is something that occasionally appears as a result of statistical sampling. However, there is no impact to the 80% prediction bounds, because that particular curve is literally "off the chart" and outside the 10% to 90% range bootstrap estimates.

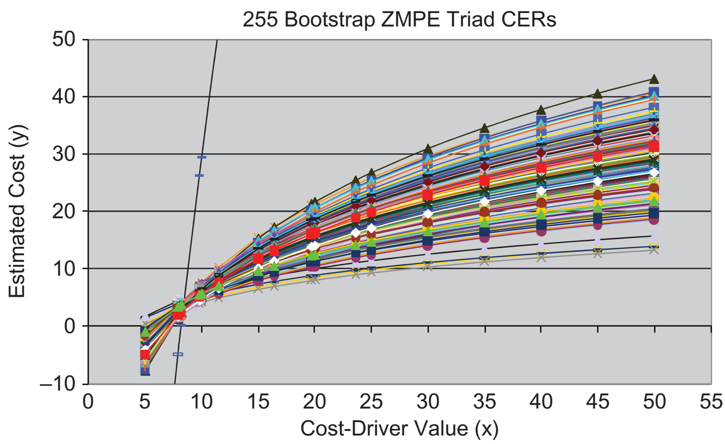

FIGURE 15 Two hundred fifty-five ZMPE bootstrap power CERs (color figure available online).

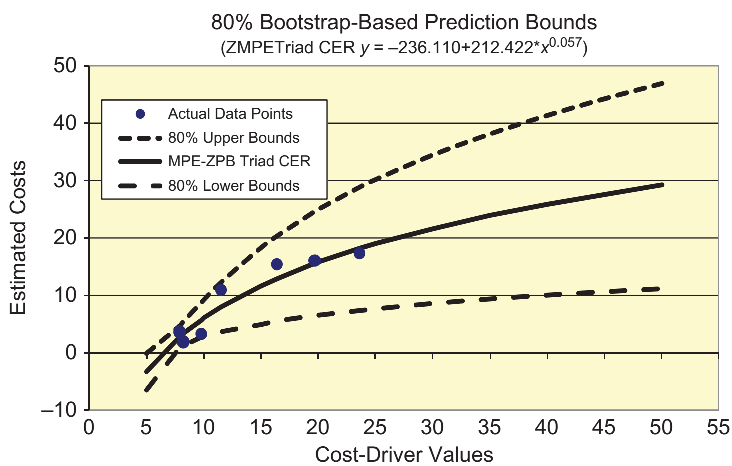

FIGURE 16 Approximate 80% upper and lower prediction bounds on estimates based on the ZMPE triad CER (color figure available online).

Summary

While explicit formulas exist for prediction intervals that correspond to OLS-derived linear CERs, those intervals can be reproduced fairly well by adjusting bounds derived by bootstrap sampling. For general-error CERs, however, prediction bounds do not appear to be available, although techniques for deriving such CERs have been available for almost 40 years. While awaiting the discovery of the "exact" solution, the bootstrap sampling technique appears to offer an opportunity to approximate prediction bounds for any specific CER by analogy with the adjustment process for OLS BBs. The bootstrap method is, however, at a significant disadvantage when compared with algebraic formulas such as those for OLS CERs: it provides a solution on a "one-time-only" basis for each CER, rather than via an algebraic formula of wide applicability. Theoretical research in this direction may be worthwhile, as it may lead to future algebraic solutions, and should be encouraged, but there is a current need to have the ability to at least approximate prediction bounds for general-error CERs.

Acronyms

B-B	Bootstrap-Based
B-BB	Bootstrap-Based Bound
BB	Bootstrap Bound
CER	Cost-Estimating Relationship
EST	Estimated
FY	Fiscal Year
IRLS	Iteratively Reweighted Least Squares
K	Thousands (usually of dollars)
MPE-ZPB	Minimum Percentage Error–Zero Percentage Bias
OLS	Ordinary Least Squares
SEE	Standard Error of the Estimate
ZMPE	Zero Percentage Bias–Minimum Percentage Error

TABLE 24 Path from ZMPE triad CER residuals to bootstrap ZMPE triad CERs

	Bootstrap *y* values						
x Values	#1	#2	#3	#4	#5	#6	...
7.9	3.331	3.826	3.595	2.657	3.331	2.657	...
8.2	1.900	4.248	4.248	4.248	3.411	1.896	...
9.8	6.850	7.868	7.868	3.306	5.464	3.300	...
11.5	4.580	8.223	8.223	7.570	10.900	10.900	...
16.4	16.656	13.373	12.312	7.449	13.373	12.312	...
19.7	16.074	8.954	18.552	18.552	20.020	14.798	...
23.6	10.451	18.763	10.451	23.369	10.432	17.274	...
a (intercept)	−234.930	−233.629	−234.175	−235.904	−236.560	−236.531	...
b (coefficient)	213.612	215.212	214.377	212.627	211.969	211.998	...
c (exponent)	0.051	0.048	0.050	0.056	0.059	0.057	...
% std error	43.87%	23.68%	26.03%	38.48%	30.23%	35.10%	...
% bias	0.0000%	0.0000%	0.0000%	0.0000%	0.0000%	0.0000%	...
R^2	91.45%	91.46%	91.45%	91.43%	91.42%	91.42%	...

Acknowledgments

The author would like to thank Erik Burgess of Burgess Consulting, Inc., and Tim Anderson of The Aerospace Corporation for their advice and assistance, as well as Steve Rozzi of Quaternion Consulting, Inc., Don MacKenzie and Dr. Mitch Robinson of Wyle Labs, and Nathan Menton, formerly with MCR and now with the United Nations in New York, for their valuable comments and suggestions during the course of this work. Also acknowledged are the valuable comments of MCR's internal technical reviewers of this article—Dr. Neal Hulkower, Amanda J. Feather, and Christopher Metz—whose work led to the correction of several typos and other defects in the article.

References

Book, S. (2006a, February 14–17). Prediction Bounds for General-Error-Regression CERs. *39th DoDCAS*, Williamsburg VA.

Book, S. (2006b). Unbiased Percentage-Error CERs with Smaller Standard Errors. *The Journal of Cost Analysis & Management*, 8(1), 55–71.

Book, S. (2010, June 8–11). Multiplicative-Error Regression. Practitioner Training Track, *2010 ISPA/SCEA Joint Annual Conference & Training Workshop*, San Diego, CA (Also presented to six previous ISPA/SCEA Annual Conferences and the European Aerospace Working Group on Cost Engineering (EACE), Cranfield University, Milton Keynes/Bedford, UK, October 19–21, 2004).

Book, S., & Lao, N. (1999, November). Minimum-Percentage-Error Regression under Zero-Bias Constraints. In *Proceedings of the Fourth Annual U.S. Army Conference on Applied Statistics, 21–23 October 1998*, U.S. Army Research Laboratory Report No. ARL-SR-84 (pp. 47–56).

Book, S., & Young, P. (1997). General-Error Regression for Deriving Cost-Estimating Relationships. *The Journal of Cost Analysi*s, 14(2), 1–28.

Book, S., & Young, P. (2006). The Trouble with R^2. *Journal of Parametrics*, 25(1), 87–114.

Casella, G. (2003). Introduction to the Silver Anniversary of the Bootstrap. *Statistical Science*, 18(2), 133–134.

Deflandre, D., and Kleijnen, J. P. C. (2002, June). *Statistical analysis of random simulations: Bootstrap tutorial*. Center for Economic Research, Tilburg University, The Netherlands (Discussion Paper No. 2002-58). Retrieved from http://greywww.kub.nl:2080/greyfiles/center/2002/doc/58.pdf

Efron, B. (1979). The 1977 Rietz Lecture—Bootstrap Methods; Another Look at the Jacknife. *Annals of Statistics*, 7, 1–26.

Efron, B. (2003). Second Thoughts on the Bootstrap. *Statistical Science*, 18(2), 135–140.

Eskew, H. L., & Lawler, K. S. (1994). Correct and Incorrect Error Specifications in Statistical Cost Models. *Journal of Cost Analysis*, 107.

Gauss, C. F. (1809). *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientum*.

Gauss, C. F. (1821/1823). Theoria Combinationis Observationum Erroribus Minimis Obnoxiae.

Goldberg, M. S., & Tuow, A. E. (2003). *Statistical Methods for Learning Curves and Cost Analysis*. Institute for Operations Research and Management Science (INFORMS).

Hu, S.-P. (2010). R^2 vs. r^2. *Journal of Cost Analysis and Parametrics*, 3(2), 13–27.

Jørgensen, B. (1997). *The Theory of Dispersion Models*. London: Chapman & Hall.

Léger, C., Politis, D. N., and Romano, J. P. (1992, November). Bootstrap Technology and Applications. *Technometrics*, 34, 378–398.

Nguyen, P., Lozzi, N., Book, S., et al. (1994). *Unmanned Space Vehicle Cost Model (USCM7)*. USAF Space and Missile Systems Center.

Nguyen, P.; Lozzi, N.; et al. (2004). *Unmanned Space Vehicle Cost Model (USCM8)*. USAF Space and Missile Systems Center.

Sperling, R., & Goldberg, M. (n.d.). Quantifying Uncertainty of Predictions from an Intrinsically Nonlinear Cost Estimation Relationship. In Preparation.

Wedderburn, R. W. M. (1974). Quasi-Likelihood Functions, Generalized Linear Models, and the Gauss-Newton Method. *Biometrika*, 61(3), 439–447.

About the Author

Dr. Stephen A. Book vacated the position of Chief Technical Officer of MCR, LLC in 2010 (after serving in that position for almost a decade) to concentrate on research, training, and subject-matter-expert customer support. In his former capacity, he was responsible for ensuring technical excellence of MCR products, services, and processes by encouraging process improvement, maintaining quality control, and training employees and customers in cost and schedule analysis and associated program-control disciplines. Earlier, at The Aerospace Corporation, he was a principal contributor to several Air Force cost studies of national significance, including the DSP/FEWS/BSTS/AWS/Brilliant Eyes Sensor Integration Study (1992) and the ALS/Spacelifter/EELV Launch Options Study (1993). He served on national panels as an independent reviewer of NASA programs, for example the 2005 Senior External Review Team on cost-estimating methods for the Exploration Systems Mission Directorate, the 1997–98 Cost Assessment and Validation Task Force on the International Space Station (Chabrow Committee), and the 1998–99 National Research Council Committee on Space Shuttle Upgrades. Dr. Book joined MCR in January 2001 after 21 years with Aerospace, where he held the title Distinguished Engineer during 1996–2000 and served as Director, Resource and Requirements Analysis Department, during 1989–1995. Dr. Book was co-editor of the ISPA/SCEA technical journal, *The Journal of Cost Analysis and Parametrics*. He received the 2010 SCEA Lifetime Achievement Award, the 2009 NASA Cost Contractor of the Year award, the 2005 ISPA Freiman Award for Lifetime Achievement, and the 1982 Aerospace Corporation Presidents Award for Analytic Achievement. Dr. Book earned his Ph.D. in mathematics, with concentration in probability and statistics, at the University of Oregon. Dr. Book passed away in January 2012.