THE IMPACT OF USING LOG-ERROR CERS OUTSIDE THE DATA RANGE AND PING FACTOR

By Dr. Shu-Ping Hu
Tecolote Research Inc.
5266 Hollister Ave. Ste. 301
Santa Barbara, CA 93111

Abstract

This paper discusses the pros and cons of using the log-error model versus the Minimum-Unbiased-Percentage Error (MUPE) model for cost estimating relationship (CER) development. Further, we discuss a very common and incorrect interpretation of the log-error CER result: it is commonly assumed to be the mean of a log-normal distribution, which is proven to be wrong. The theoretical derivations of two available correction factors (Goldberg, PING) to adjust for the mean in unit space for log-error models are discussed accordingly. We also show that the Goldberg equation outside the data range generates counter-intuitive and inappropriately low results by comparing three commonly implemented log-linear equations: the Goldberger, the PING Factor, and the uncorrected equations.

Multiplicative error terms are commonly used in the cost analysis field because experience tells us that the error of an individual observation (e.g., cost) is generally proportional to the magnitude of the observation (not a constant). The log-error model is a popular way to generate these types of CERs because ordinary least squares (OLS) can be accomplished in log space if the fitted equation is log-linear. However, the log-linear CER will be biased low when transformed back into unit space; it will predict closer to the median (not the mean) of the CER risk distribution in unit space. Therefore, correction factors are required to adjust the CER result to produce the mean in unit space.

The MUPE regression model is an alternative method to hypothesize the multiplicative error in a CER. All of the Unmanned Space Vehicle Cost Model, Eighth Edition (USCM8) CERs (Reference 5) have been developed using MUPE. The MUPE method involves an iterative, weighted least squares regression that provides unbiased percentage error regression results. No transformation or adjustment (to correct the bias in unit space) is needed for fitting a MUPE equation.

Introduction

For many CERs, the error of an individual observation (e.g., cost) is approximately proportional to the magnitude of the observation. In such cases, it is appropriate to hypothesize a multiplicative error term for the CER.

Several optimization techniques have been used to model multiplicative errors over the years. One common practice was to work in log space by taking natural logs of both the dependent variable and the equation form. When the transformed equation is linear in log space, OLS can be applied to derive a Best Linear Unbiased Estimator (BLUE) in log space, which is also the Maximum Likelihood Estimator (MLE) in log space. If the equation form is not log-linear, the log transform can still be used to model a multiplicative error, but the optimization will be non-linear

least squares. Although the mean and median are the same for the log-linear CERs in log space, when transforming the equation back to unit space the mean and median differ. The unit space CER predicts closer to the median instead of the mean. Therefore, the direct translation of the equation back to unit space tends to underestimate the mean value of the original population. A multiplicative correction factor (introduced as the "PING Factor" at Reference 6 in 1988) is used to adjust the CER result to more closely reflect the mean in unit space.

Log Error Model. The multiplicative error model is generally stated as follows:

$$Y_i = f(\mathbf{x}_i, \boldsymbol{\beta})\,\varepsilon_i \qquad \text{for i = 1, …, n} \tag{1}$$

where:

n	= sample size
Y_i	= observed cost of the ith data point, i = 1 to n
$f(\mathbf{x}_i,\boldsymbol{\beta})$	= the value of the hypothesized equation at the ith data point
$\boldsymbol{\beta}$	= vector of coefficients to be estimated by the regression equation
$\boldsymbol{x}_i$	= vector of cost driver variables at the ith data point
ε_i	= error term

If the multiplicative error term (ε_i) is further assumed to follow a log-normal distribution, then the error can be measured by the following:

$$e_i = \ln(\varepsilon_i) = \ln(Y_i) - \ln(f(\mathbf{x}_i, \boldsymbol{\beta})) \tag{2}$$

where "ln" stands for nature logarithmic function. The objective is then to minimize the sum of squared e_is (i.e., $(\Sigma(\ln(\varepsilon_i))^2$). If the transformed function is linear in log space, then OLS can be applied in log space to derive a solution for $\boldsymbol{\beta}$. If not, we need to apply the non-linear regression technique to derive a solution.

Although a least squares optimization in log space produces an unbiased estimator in log space, the estimator is no longer unbiased when transformed back to unit space (see References 1, 2, and 4). However, the magnitude of the bias can be corrected with a simple factor if the errors are distributed normally in log space (see References 1 and 2). Because of this shortcoming, the MUPE method is recommended for modeling multiplicative error directly *in unit space* to produce unbiased estimators.

The MUPE Method. The general specification for a MUPE model is the same as given above (Equation 1), except that the error term is assumed to have a mean of 1 and variance σ^2. Based upon this assumption of a multiplicative model, a generalized error term is defined by:

$$e_i = \frac{y_i - f(\mathbf{x}_i, \boldsymbol{\beta})}{f(\mathbf{x}_i, \boldsymbol{\beta})} \tag{3}$$

where e_i now has mean of 0 and variance σ^2.

This percentage error differs from the traditional percentage error in the denominator, where MUPE uses predicted cost instead of actual cost as the baseline. The optimization objective is to find the coefficient vector β that minimizes the sum of squared e_is:

$$\text{Minimize} \quad \sum_{i=1}^{n} \left(\frac{y_i - f(\mathbf{x}_i, \boldsymbol{\beta})}{f(\mathbf{x}_i, \boldsymbol{\beta})} \right)^2 = \sum_{i=1}^{n} e_i^{\,2} \tag{4}$$

Tecolote Research, Inc. has proposed a MUPE regression technique to solve for the function in the numerator separately from the function in the denominator (see References 2 and 5). This is done through an iterative process.

$$\text{Minimize} \quad \sum_{i=1}^{n} \left(\frac{y_i - f(\mathbf{x}_i, \boldsymbol{\beta}_k)}{f(\mathbf{x}_i, \boldsymbol{\beta}_{k-1})} \right)^2 = \sum_{i=1}^{n} \left(\frac{y_i - f_k(\mathbf{x}_i)}{f_{k-1}(\mathbf{x}_i)} \right)^2 \tag{5}$$

where k is the iteration number and the other terms are as defined previously.

The weighting factor of each residual in the current iteration is equal to the reciprocal of the predicted value from the previous iteration. Since the denominator in Equation (5) is kept fixed throughout the iteration process, the MUPE technique turns out to be a weighted least squares with an additive error. The final solution is derived when the change in the estimated coefficients (β vector) between the current iteration and the last iteration is within the analyst-specified tolerance limit. This optimization technique (Equation 5) is commonly referred to as Iteratively Reweighted Least Squares (IRLS; see References 8 and 9).

PROS AND CONS OF MUPE AND LOG-ERROR MODELS

Model developers must decide the best way to model the error of their equation, choosing either the MUPE model or the log-error model. Here are a few advantages of using the MUPE method:

- The MUPE CER has zero proportional error for all points in the database (no sample bias).
- No transformation or adjustment (to correct the bias in unit space) is needed for fitting a MUPE equation.
- Goodness-of-fit measures (or asymptotic goodness-of-fit measures) can be applied to judge the quality of the model under the normality assumption.
- The MUPE CER produces consistent estimates of the parameters and the mean of the equation.
- The estimated parameters using the MUPE method are also the maximum likelihood estimates (MLE) of the parameters (by Matthew Goldberg, 2001).

A disadvantage of using the MUPE model is that it relies on the non-linear optimization technique to achieve a solution, which can be cumbersome. See the detailed descriptions of the MUPE method in References 2 and 3.

The following concerns have been raised about using the log-error model (Equation 2):

1. Errors are not expressed in meaningful units (i.e., $\ln(\varepsilon_i)$s are in log of dollars).

2. Minimizing $\Sigma(\ln(\varepsilon_i))^2$ is not the same as minimizing $\Sigma(\varepsilon_i)^2$. As a result, the standard error of estimate (SEE) derived in log space for log-linear CERs cannot be compared with the SEE in unit space for non-linear regression equations.
3. We must choose power form CERs to use the log-error assumption.
4. The resultant equation will be biased low when transforming back to unit space. To obtain an unbiased estimate, we need to multiply the CER by a correction factor to estimate the mean in unit space.

Although these concerns all appear to be legitimate, we think only the last one on the list is a valid concern; the other three are **not**. We will address these concerns in the order given above. We understand that log-errors (Equation 2) are not expressed in dollars as units. However, the log-errors have an interesting interpretation: $\ln(\varepsilon_i)$ can be viewed "approximately" as a percentage error by Taylor series expansion. It actually matches the MUPE definition of percentage error:

$$\ln(\varepsilon_i) \approx \varepsilon_i - 1 = \frac{Y_i}{f(\mathbf{x}_i, \boldsymbol{\beta})} - 1 = \frac{Y_i - f(\mathbf{x}_i, \boldsymbol{\beta})}{f(\mathbf{x}_i, \boldsymbol{\beta})} \tag{6}$$

Although minimizing $\Sigma(\ln(\varepsilon_i))^2$ is not the same as minimizing $\Sigma(\varepsilon_i)^2$, this is not a problem because we hypothesize a multiplicative model (Equation 2), not an additive one. As a general rule, the measure of SEE in fit space (a fit measure) cannot be compared across different models. We should not compare the fit measures between two models if they are developed using different fit criteria. For example, comparing the SEEs between an additive model and a multiplicative model is meaningless because one model minimizes the sum of squared *absolute* errors while the other minimizes the sum of squared *percentage* errors.

Now let us address the third concern on the list. Of course, we have to choose "log-linear" equations if we want to apply OLS to fit equations in log space. However, we are not forced to use OLS. To model the cost variation, we should (1) hypothesize a CER form based upon good logic and solid technical grounds and (2) choose an appropriate error term assumption. Note that the choice of the CER form should not drive the error term assumption and vice versa. If the regression equation cannot be solved linearly, we will apply the non-linear optimization technique to generate a solution. For example, if a fixed cost term is guided by the engineering judgment and errors are assumed to be scaled with the project, we can certainly hypothesize this equation form, $a + bX^c$, with a multiplicative error term to explain the cost variation. In this case, we need to use the non-linear regression technique to derive a solution.

The last one on the list has been a very common concern for years. Although a least squares solution is unbiased in log space, the resultant CER is no longer unbiased when transforming back to unit space. This is the most important reason why the MUPE method is suggested for modeling multiplicative errors in unit space (to avoid the use of correction factors to adjust for the mean).

Despite the valid concern that the resultant equation is biased low in unit space, the log-error model is still quite popular when modeling multiplicative errors. The reasons are given below:

- If the hypothesized equation is log-linear, e.g., $y = ax^b\varepsilon$, then the regression can be done in log space linearly, which does not involve an iterative process. In this case, we have the following advantages:
  - The traditional goodness-of-fit measures can be applied to judge the quality of the fit in log space.
  - The outliers can be easily identified for further scrutiny.
  - The prediction intervals can also be easily derived.

  Obviously, the above advantages are all restricted to "log-linear" models. They are no longer valid for non-linear CERs, such as $y = (ax^b + c)\varepsilon$.
- The standard error of estimate in log space (SEE_L) can be regarded as a measure of a percentage error at a certain given x level in unit space, i.e., $SEE_L \cong CoV_A$ at a given x level. Note that CoV_A denotes the coefficient of variation in unit space expressed as a percentage. This approximation is derived by applying Taylor series expansion to the ratio of Equation 14 over Equation 12. See Reference (1) for details.
- The log-error can be viewed approximately as the MUPE percentage error (see Equation 6).

Based upon the above discussions, it appears that the MUPE method is superior to the log-error model. In addition, its error term assumption is more generic. Nonetheless, the choices between the MUPE and log-error models should depend upon the error term assumption:

- Choose MUPE if the error term (ε) is associated with a mean of one and variance of σ^2.
- Choose log-error model if ε follows a log-normal distribution with a mean of zero and variance σ^2 in log space, i.e., $\varepsilon \sim LN(0, \sigma^2)$.

The real purpose of using a CER is for predicting a future cost. The impact of applying correction factors to log-error CERs might be more severe than we originally thought when the prediction is outside the database. We will begin the discussion by introducing the correction factors and the related theory.

DERIVATION OF CORRECTION FACTORS

Theories for Log-Linear Models. Let us hypothesize a log-linear equation with a multiplicative error term as given in Equation 1:

$$Y_i = f(\mathbf{x_i}, \boldsymbol{\beta})\varepsilon_i = e^{\beta_0} x_{1i}^{\beta_1} x_{2i}^{\beta_2} \cdots x_{ki}^{\beta_k} \varepsilon_i \quad \text{for i} = 1,...,n \tag{7}$$

where:

ε_i's are independently, identically distributed (i.i.d.) random variables associated with a log-normal distribution with mean of 0 and variance σ^2 in log space, i.e., $LN(0,\sigma^2)$,
$\beta_o, \beta_1, ..., \beta_k$, and σ^2 are unknown parameters,
$x_{1i}, x_{2i}, ..., x_{ki}$ are the independent variables for the i^{th} data point, and
k is the total number of independent variables in the model.

The above model can be equivalently stated as

$$\ln(Y_i) = \beta_0 + \sum_{j=1}^{k} \beta_j \ln(x_{ji}) + \ln(\varepsilon_i) \quad \text{for i} = 1,...,n \tag{8}$$

The dependent variable in log space ($\ln(Y_i)$) now follows a normal distribution because $\ln(\varepsilon_i)$ is distributed as $N(0,\sigma^2)$. Therefore, at a given x value, say $\mathbf{x} = \mathbf{x_o} = (x_{1o}, x_{2o}. . . , x_{ko})$, the conditional distributions of Y in both *log* and *unit* space are given by Equation 9 and Equation 10, respectively:

$$\ln(Y / \mathbf{x} = \mathbf{x_o}) = \beta_0 + \sum_{j=1}^{k} \beta_j \ln(x_{jo}) + \ln(\varepsilon) = \mu_o + \ln(\varepsilon) \sim N(\mu_o, \sigma^2) \tag{9}$$

$$(Y / \mathbf{x} = \mathbf{x_o}) = \exp(\mu_o + \ln(\varepsilon)) \sim LN(\mu_o, \sigma^2) \tag{10}$$

Where $\mu_o = \beta_o + \Sigma\, \beta_j \ln(x_{jo})$

It follows from Equation 9 that the conditional mean and median value of Y (in log space) at this given value, $\mathbf{x_o}$, are both equal to μ_o:

$$E(\ln(Y / \mathbf{x} = \mathbf{x_o})) \quad = \beta_0 + \sum_{j=1}^{k} \beta_j \ln(x_{jo}) = \mu_o \tag{11}$$

However, it can be easily shown by Equation 10 that the conditional mean, median, and standard deviation of Y (at the given $\mathbf{x_o}$) in unit space are given respectively by the following equations:

$$E(Y / \mathbf{x} = \mathbf{x_o}) \quad = \exp(\mu_o + \sigma^2/2) = e^{\beta_0} x_{1o}^{\beta_1} x_{2o}^{\beta_2} \ldots x_{ko}^{\beta_k} (e^{\sigma^2/2}) = \mu_A \tag{12}$$

$$Meidan(Y / \mathbf{x} = \mathbf{x_o}) \quad = \exp(\mu_o) = e^{\beta_0} x_{1o}^{\beta_1} x_{2o}^{\beta_2} \ldots x_{ko}^{\beta_k} = M_A \tag{13}$$

$$Stdev(Y / \mathbf{x} = \mathbf{x_o}) \quad = \exp(\mu_o + \sigma^2/2)\sqrt{e^{\sigma^2} - 1} = e^{\beta_0} x_{1o}^{\beta_1} x_{2o}^{\beta_2} \ldots x_{ko}^{\beta_k} (e^{\sigma^2/2})\sqrt{e^{\sigma^2} - 1} \tag{14}$$

(Note: If a random variable Y is distributed as $LN(\mu, \sigma^2)$, then $E(Y) = \exp(\mu+\sigma^2/2)$, $Median(Y) = \exp(\mu)$, and $Var(Y) = (E(Y))^2(\exp(\sigma^2)-1)$.) Furthermore, the mode of Y at the given $\mathbf{x_o}$ is neither the mean nor the median:

$$Mode(Y / \mathbf{x} = \mathbf{x_o}) \quad = \exp(\mu_o - \sigma^2) = e^{\beta_0} x_{1o}^{\beta_1} x_{2o}^{\beta_2} \ldots x_{ko}^{\beta_k} e^{-\sigma^2} = Mode_A \tag{15}$$

Therefore, the term $e^{\sigma^2/2}$ can be regarded as a factor explaining the difference between μ_A and M_A:

$$\boxed{\mu_A / M_A = \exp(\sigma^2/2)} \tag{16}$$

It is clear from Equation 16 that the direct translation of the solution from log space to unit space is not a good estimator of the population mean, μ_A, if the difference between mean (μ_A) and median (M_A) is not negligible. Therefore, we have the following conclusion:

> The log-linear equation will be biased low if we do not apply correction factors.

Procedure. In the following paragraphs, we will illustrate the procedure of developing correction factors. We will also prove (1) the log-linear CER in unit space (denoted by $\hat{Y}$) follows a log-normal distribution (see Equation 19) and (2) the variance of this predictor depends on the location of the driver variables.

To compute the mean and standard deviation of $\hat{Y}$, let us first introduce a variable r_o:

$$r_o = \ln(\mathbf{x_o})(X'X)^{-1} \ln(\mathbf{x_o})^t \quad (17)$$

where:

$\ln(\mathbf{x_o})$ = $(1, \ln(x_{1o}), ..., \ln(x_{ko}))$, a row vector of given driver values in log space and 1 is for the intercept

X = design matrix in log space

(The superscript t denotes the transpose of a vector or a matrix.)

For simplicity, the letter X is also used to denote the data matrix in log space. Note that r_o is the so-called leverage value in log space if $\mathbf{x_o}$ is a vector of independent variables in the data matrix. Both the mean and standard deviation of $\hat{Y}$ (at the given $\mathbf{x_o}$) are functions of r_o. To be more specific, the standard deviation of a future prediction in log space is equal to $r_o\sigma^2$ (see Equation 18). In a one-independent variable case, the value of r_o is given by

$$r_o = (1, \ln(x_o))(X'X)^{-1}\begin{pmatrix} 1 \\ \ln(x_o) \end{pmatrix} = \frac{1}{n} + \frac{(\ln(x_o) - \overline{\ln(x)})^2}{\sum_i (\ln(x_i) - \overline{\ln(x)})^2} = \frac{1}{n} + \frac{(\ln(x_o) - \overline{\ln(x)})^2}{SSx}$$

It is clear that the variable r_o captures the sample size and the distance of the estimating point from the center of the database in terms of the sample standard deviation of the driver variable.

If the classical assumptions hold as explained in Equation 7, the OLS method can generate an unbiased estimator for the dependent variable <u>in log space</u>. This OLS predictor of Y in log space (at the given $\mathbf{x_o}$) follows a normal distribution because it is a linear combination of the normally distributed $\ln(Y_1)$, $\ln(Y_2)$,..., and $\ln(Y_n)$:

$$\ln(\hat{Y} / \mathbf{x} = \mathbf{x_o}) = \ln(\mathbf{x_o})\hat{\boldsymbol{\beta}} \sim N(\ln(\mathbf{x_o})\boldsymbol{\beta},\ r_o\sigma^2) = N(\mu_o,\ r_o\sigma^2) \quad (18)$$

where:

$\boldsymbol{\beta}$ = $(\beta_o, \beta_1, ..., \beta_k)^t$, a column vector of the unknown coefficients

$\hat{\boldsymbol{\beta}}$ = $(X'X)^{-1}X'(\ln(\mathbf{Y}))$, the OLS solution for $\boldsymbol{\beta}$

$\ln(\mathbf{Y})$ = $(\ln(Y_1), \ln(Y_2),..., \ln(Y_n))^t$, a column vector of $\mathbf{Y}$ in log space

$\ln(\mathbf{x_o})$ = $(1, \ln(x_{1o}), ..., \ln(x_{ko}))$, $\mu_o = \ln(\mathbf{x_o})\boldsymbol{\beta}$, and r_o are all given above

Note that the mean and variance of $\ln(\hat{Y} / \mathbf{x} = \mathbf{x_o})$ are derived by matrix algebra (see Reference 9) and clearly the variance of this predictor is driven by the location of the data point.

By definition, therefore, the distribution of the direct translation of Equation 18 into unit space follows a log-normal distribution:

$$(\hat{Y} / \mathbf{x} = \mathbf{x_o}) \quad = \exp(\ln(\mathbf{x_o})\hat{\boldsymbol{\beta}}) = e^{\hat{\beta}_0} x_{1o}^{\hat{\beta}_1} x_{2o}^{\hat{\beta}_2} \ldots x_{ko}^{\hat{\beta}_k} \sim LN(\mu_o, r_o\sigma^2) \tag{19}$$

Based upon Equations 19, the mean, median, and standard deviation of $\hat{Y}$, at a given $\mathbf{x_o}$ vector, are given by Equations 20, 21, and 22, respectively:

$$E(\hat{Y} / \mathbf{x} = \mathbf{x_o}) \quad = \exp(\mu_o + r_o\sigma^2 / 2) \tag{20}$$

$$Meidan(\hat{Y} / \mathbf{x} = \mathbf{x_o}) \quad = \exp(\mu_o) \tag{21}$$

$$Stdev(\hat{Y} / \mathbf{x} = \mathbf{x_o}) \quad = \exp(\mu_o + r_o\sigma^2 / 2)\sqrt{\exp(r_o\sigma^2) - 1} \tag{22}$$

And the net correction factor for estimating the mean value in unit space at the given $\mathbf{x_o}$ vector is the ratio between Equation 12 and Equation 20:

$$\boxed{\text{Net CF for Mean at } \mathbf{x_o}: \quad E(Y / \mathbf{x} = \mathbf{x_o}) \big/ E(\hat{Y} / \mathbf{x} = \mathbf{x_o}) \quad = \exp((1 - r_o)\frac{\sigma^2}{2})} \tag{23}$$

Illustrations of Simple Cases. Let us use two simple examples to illustrate Equation 23. In the univariate case, where $Y_1, Y_2, \ldots, Y_n$ are i.i.d. random variables associated with $LN(\mu,\sigma^2)$, the estimator of the mean is equal to $\exp(\overline{\ln(Y)})$. The distributions of $\overline{\ln(Y)}$ and $\exp(\overline{\ln(Y)})$ are given by Equation 24 and Equation 25, respectively:

$$\overline{\ln(Y)} = \sum_{i=1}^{n} \ln(Y_i) / n \quad \sim N(\mu, \sigma^2/n) \tag{24}$$

$$\exp(\overline{\ln(Y)}) = \sqrt[n]{\Pi Y_i} \quad \sim LN(\mu, \sigma^2/n) \tag{25}$$

It follows from Equation 25 that $E(\exp(\overline{\ln(Y)})) = \exp(\mu + \sigma^2/2n)$. Thus, for predicting the population mean (i.e., $\exp(\mu + \sigma^2/2)$), the net correction factor for $\hat{Y}$ (i.e., $\exp(\overline{\ln(Y)})$) can be expressed as

$$\exp(\sigma^2/2) / \exp(\sigma^2/2n) = \exp((1 - 1/n)\sigma^2/2) \tag{26}$$

The first term in Equation 26 indicates that the mean is underestimated by $\exp(\sigma^2/2)$, which can be regarded as a transformation bias. The second term in Equation 25 indicates that the median is overestimated by $\exp(\sigma^2/2n)$, which can be regarded as a sampling bias.

For the one independent variable model when $x = x_o$, the predicted values of Y in log space and unit space are given by Equations 27 and 28, respectively:

$$\ln(\hat{Y} / x = x_o) = (1, \ln(x_o))\begin{pmatrix} \hat{\beta}_o \\ \hat{\beta}_1 \end{pmatrix} = \hat{\beta}_o + \hat{\beta}_1 \ln(x_o) \sim N(\beta_o + \beta_1 \ln(x_o), r_o\sigma^2) \tag{27}$$

$$(\hat{Y} / x = x_o) = \exp(\hat{\beta}_o + \hat{\beta}_1 \ln(x_o)) = e^{\hat{\beta}_o} x_o^{\hat{\beta}_1} \sim LN(\beta_o + \beta_1 \ln(x_o), r_o \sigma^2) \quad (28)$$

where:

$$r_o = (1, \ln(x_o))(X'X)^{-1}\begin{pmatrix} 1 \\ \ln(x_o) \end{pmatrix} = \frac{1}{n} + \frac{(\ln(x_o) - \overline{\ln(x)})^2}{SSx}, \quad (29)$$

β_o and β_1 are the unknown coefficients; $\hat{\beta}_o$ and $\hat{\beta}_1$ are their estimated coefficients,
SSx is the sum of squared deviation of the driver variable about its mean in log space,
$\overline{\ln(x)}$ is the average of the independent variable in log space, and
x_o is a given x value.

The expected value of the prediction when $x = x_o$ is then given by

$$\begin{aligned} E(\hat{Y} / x = x_o) &= \exp(\beta_o + \beta_1 \ln(x_o) + r_o \sigma^2 / 2) \\ &= e^{\beta_o} x_o^{\beta_1} \exp\left((\frac{1}{n} + \frac{(\ln(x_o) - \overline{\ln(x)})^2}{SSx}) \frac{\sigma^2}{2} \right) \end{aligned} \quad (30)$$

When Equation 30 is compared to Equation 12, the net correction factor for $\hat{Y}$ at the given x_o is

$$\exp\left((1 - r_o) \frac{\sigma^2}{2} \right) = \exp\left((1 - \frac{1}{n} - \frac{(\ln(x_o) - \overline{\ln(x)})^2}{SSx}) \frac{\sigma^2}{2} \right) \quad (31)$$

Unbiased Correction Factor (Goldberger's Factor). In the following section, we will demonstrate that the net correction factor, Goldberger's correction factor (GF) can be expressed as

$$\boxed{GF = g\left((1 - r_o) s^2 / 2\right) \cong \exp\left((1 - r_o) s^2 / 2\right)}$$

The function g is defined below (see Equation 33). Let us use s to denote "standard error of estimate" in log space. Although the estimator of the population variance σ^2 (in log space) is s^2, the statistic $\exp(s^2/2)$, on average, tends to overestimate the unknown factor $\exp(\sigma^2/2)$:

$$E(e^{s^2/2}) \geq e^{(E(s^2/2))} = e^{(\sigma^2/2)} \quad (32)$$

Note that the inequality in Equation 31 is based upon Jensen's inequality. Hence, it is necessary to develop a function to generate an unbiased estimator for the net correction factor given in Equation 22. Such a function, g, is given below:

$$g(t) = 1 + \frac{t}{1!} + \frac{(n-p)t^2}{2!(n-p+2)} + \frac{(n-p)^2 t^3}{3!(n-p+2)(n-p+4)} + \ldots \quad (33)$$

where p is the total number of coefficients to be estimated and n is the sample size.

The above-defined function g has the following property:

$$E(g(a\,s^2)) = e^{a\sigma^2} \quad \text{for any real number } a \tag{34}$$

(See Reference 1 for math derivations of Equation 34.) Hence, Goldberger suggests that we use Equations 35 and 36, respectively, as the unbiased estimators for the mean and median in unit space:

$$\textit{Mean Estimator} = e^{\hat{\beta}_0}\, x_{1o}^{\hat{\beta}_1}\, x_{2o}^{\hat{\beta}_2} \cdots x_{ko}^{\hat{\beta}_k}\; g((1-\mathrm{r_o})\,s^2/2) \tag{35}$$

$$\textit{Median Estimator} = e^{\hat{\beta}_0}\, x_{1o}^{\hat{\beta}_1}\, x_{2o}^{\hat{\beta}_2} \cdots x_{ko}^{\hat{\beta}_k}\; g(-\mathrm{r_o}\,s^2/2) \tag{36}$$

The last term in Equation 35, which is to be multiplied to the CER, is the so-called *net correction factor for the mean* or Goldberger's correction factor (GF). For simplicity, this factor can be approximated by the following term:

$$GF = g\left((1-\mathrm{r_o})\,s^2/2\right) \;\cong \exp\left((1-\mathrm{r_o})\,s^2/2\right) \tag{37}$$

This theoretical correction factor (Equation 37) suggested by Goldberger is a variable factor. It should be evaluated point by point and multiplied to the log-error CERs to obtain the theoretical mean in unit space (see Equation 35). This process is tedious and can get very cumbersome when more independent variables are introduced into the CER.

The PING Factor. The PING Factor (PF) is given by

$$\boxed{PF = g((1-\frac{p}{n})\frac{s^2}{2}) \;\cong \exp((1-\frac{p}{n})\frac{s^2}{2})}$$

Since r_o (in Equation 37) has to be evaluated at each different x level, Equations 35 and 36 are of little realistic use. We suggest using "p/n" as an approximation of r_o for any given x value, where p is the total number of estimated coefficients and n is the sample size. This way, the correction factor for the mean (to be multiplied to the CER) can be simplified as given below:

$$g((1-\frac{p}{n})\frac{s^2}{2}) \tag{38}$$

Equation 38 is commonly referred to as the PING Factor. The term p/n in the equation above is the expected value of r_o. In other words, we use the mean leverage value to approximate r_o as if $\mathbf{x_o}$ is sampled randomly from the sample population as $\mathbf{x_i}$'s in the log transformed data matrix. The proof is given by evaluating the value of r_o in the "Hat" matrix.

Let X denote the design matrix in log space as given in Equation 17. If $\mathbf{x_o}$ is a column vector in the log space data matrix, then r_o is simply the corresponding diagonal element of the n by n symmetrical "Hat" matrix H (= $X(X'X)^{-1}X'$). Since H is a square, idempotent matrix (i.e., H*H =

H), the trace of H is equal to its rank, which is the number of estimated coefficients. (The trace of a square matrix, by definition, is the sum of its diagonal elements.) Therefore, the mean value of r_o is the mean value of diagonal elements of the Hat matrix H, i.e., the number of estimated coefficients divided by the sample size.

It is clear that the PING Factor is a general correction for the level of the function; it is evaluated within the range of the database. Compared to Goldberger's Factor, the PING Factor is a handy, **constant** correction factor for the entire equation. In most situations (when n gets sufficiently large and s is moderately small, say < 0.8), the PING Factor can be further approximated by

$$PF \cong e^{\left((1-\frac{p}{n})\frac{s^2}{2}\right)} = \exp((1-\frac{p}{n})\frac{s^2}{2}) \quad (39)$$

This simplified PING Factor (Equation 39) is a good approximation of the theoretical one (Equation 38) for most cases (see Reference 2 for details). However, the difference between Equations 38 and 39 can be one percent or more if the sample size is small and/or standard error of estimate is fairly large. For example, if n = 6, p = 2, and s = 0.95, then the simplified PING Factor is 1.35 while the theoretical PING Factor is 1.33. In most applications with moderate standard error of estimates, we recommend using the simplified PING Factor.

Just as Goldberger's Factor, the first term in the PING Factor is used to adjust the downward bias between the mean and the median, which can be regarded as a transformation bias. As for the second term in Equation 39, it is used to adjust the upward bias for estimating the median. This bias can be regarded as a sampling bias because it vanishes as the sample size approaches infinity. Some analysts wonder why the PING Factor gets bigger when the sample size increases while the values of s and p remain fixed (see Equation 39). This is because the sampling bias (for estimating the median) gets smaller for a larger sample. We only need to adjust the transformation bias between the mean and median when the sample size approaches infinity or gets sufficiently large.

SENSITIVITY ANALYSIS

Now let us examine the value of r_o in Equation 37. As defined in Equation 17, it is given by

$$r_o = \ln(\mathbf{x_o})(X'X)^{-1} \ln(\mathbf{x_o})^t$$

This value should be greater than or equal to zero because r_o is part of the variance of the predictor at a given $\mathbf{x_o}$ in log space (see Equation 18). It is also bounded above by one if $\mathbf{x_o}$ is a vector in the data matrix. However, this restriction ($r_o \le 1$) no longer exists if $\mathbf{x_o}$ is not within the data range. In other words, the value of r_o can be **larger than one** if it is evaluated outside the data range. In this case, Goldberger's correction factor (i.e., $g((1- r_o)s^2/2)$ or $\exp((1-r_o)s^2/2)$ is **less than one**. When this happens, Goldberger's equation produces an estimate that is actually smaller than the uncorrected equation! As shown by Equation 31, the further the driver variable moves away from the center of the database, the smaller (in magnitude) the correction factor becomes due to increased leverage value. In learning curve analysis, the "unbiased" first unit cost

(i.e., $e^{\hat{\beta}_o} e^{(1-r_o)s^2/2}$) may be even less than **1%** of the "uncorrected" first unit cost (i.e., $e^{\hat{\beta}_o}$) due to the huge downward correction of the median for the first unit cost (i.e., T_1). This kind of unbiased T_1, as well as many other unbiased estimates for predictions outside the data range, are suspiciously low and hence should not be considered useful.

In mathematical terms, the ratio between the theoretical unbiased Goldberger's Factor (Equation 37) and the PING Factor (Equation 38) is given by

$$g((1-r_o)\frac{s^2}{2}) \Big/ g((1-\frac{p}{n})\frac{s^2}{2}) \cong \exp\left((\frac{p}{n}-r_o)\frac{s^2}{2}\right) \quad (40)$$

As explained above, this ratio can be substantially different from one if r_o is evaluated outside the data range. Let us examine whether the PING Factor is a good substitute for the Goldberger Factor for the points within the data range. A one-independent-variable model will be used as an illustrative example. Goldberger's Factor reaches its maximum at the **center** of the database in log space, and it decreases when moving away from that point. As given in Equation 31, when the distance between the independent variable and the center is within one sample standard deviation (evaluated in log space), Goldberger's equation is higher than the equation multiplied by the Ping Factor (i.e., the PING Factor equation). Goldberger's equation lies below the Ping Factor equation when the independent variable moves away from this range.

For the purpose of illustration, we chose a power equation with eight data points and a fairly large standard error of estimate (0.5). We noticed the average difference between the Goldberger and PING Factor equations is about 1% for the points within the data range, increasing to a couple of percent towards the boundaries of the data points. However, when the independent variable deviates about 30% beyond the boundaries, there is a cross-over between Goldberger's equation and the uncorrected equation. If the standard error of estimate is 0.5 or larger, Goldberger's equation decreases very rapidly when the independent variable moves further and further away from the boundaries, which makes the projection quite uncertain. We illustrate three equations in the graph in Figure 1: the unbiased Goldberger's equation, the PING Factor equation, and the uncorrected equation (with no correction factors applied). Figure 2 illustrates these equations when 0.8 is chosen as the standard error of estimate and the driver variable has a smaller exponent than that of Figure 1.

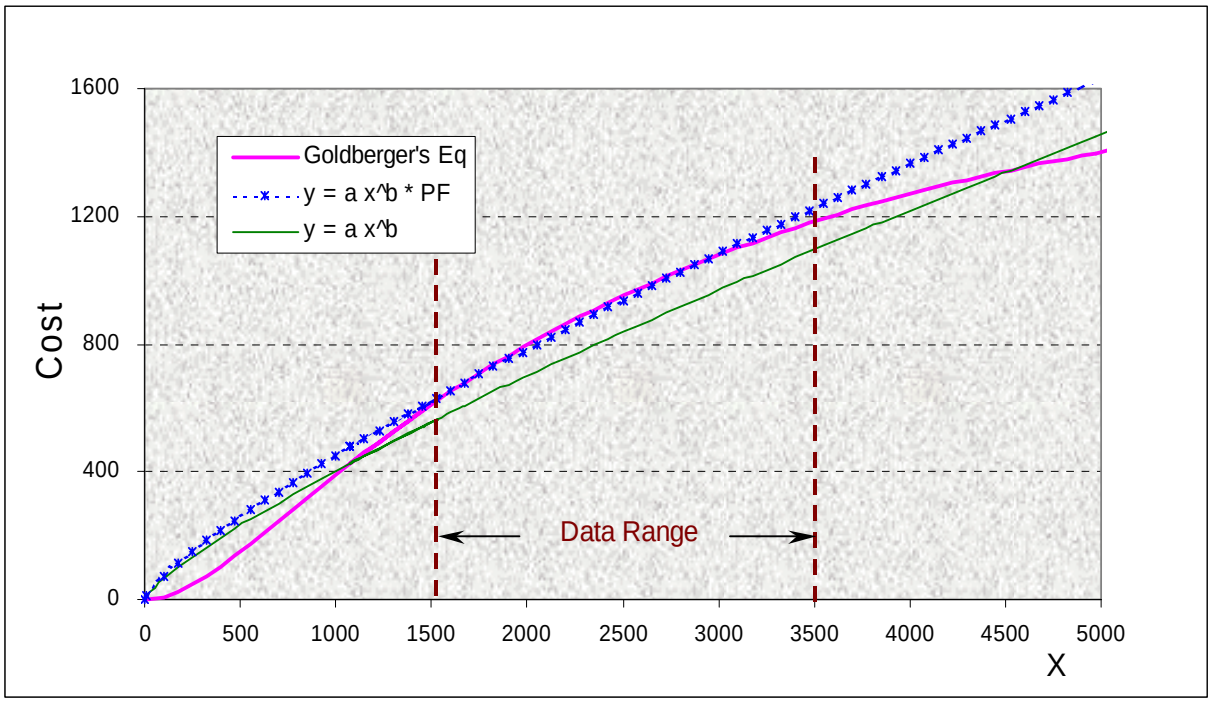

Figure 1: Comparing a Log-Linear CER with the Goldberger and PING Factor Equations Using 0.5 as the Standard Error in Log Space

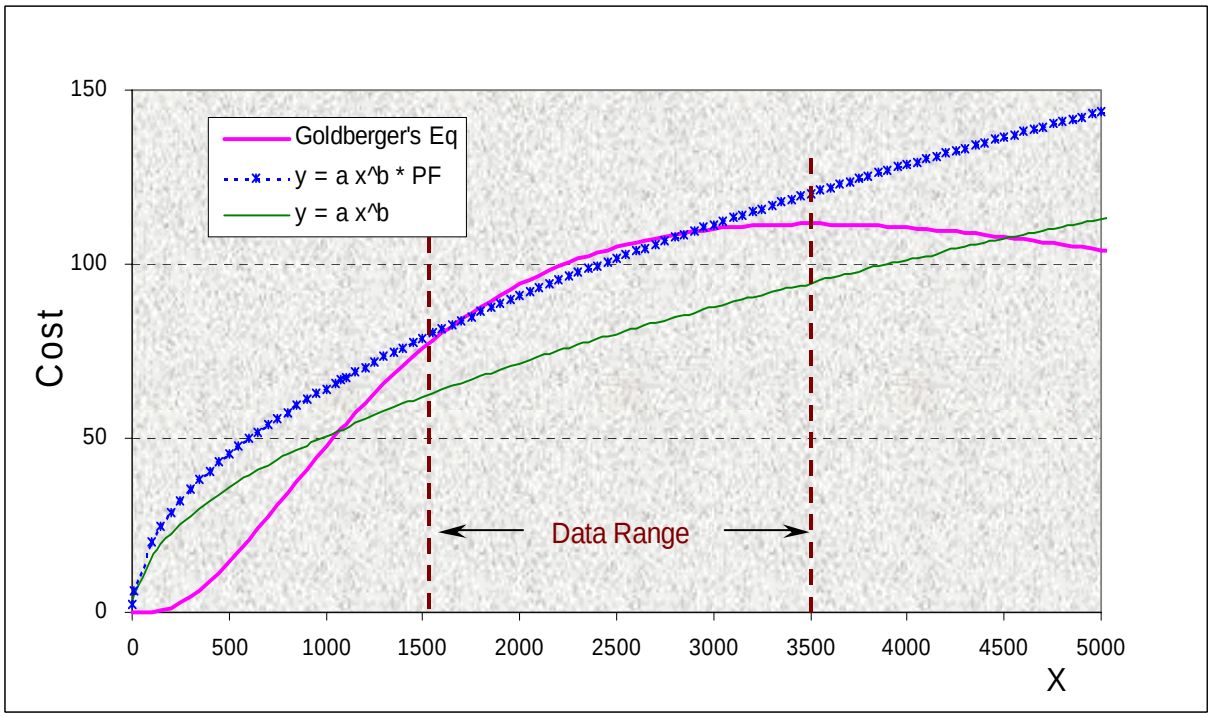

Figure 2: Comparing a Log-Linear CER with the Goldberger and PING Factor Equations Using 0.8 as the Standard Error in Log Space

Based upon the above graph, if the independent variable represents weight in pounds, then the cost of a 5000-pound "box" is cheaper than the cost of a 3000-pound "box" using the Goldberger equation. This is certainly counter-intuitive and very doubtful.

CAUTIONARY NOTES IN PREDICTION

Some references consider $\exp(\beta_0)$ to be representative of the level of the (conditional) median for the entire function. This could be very misleading if the intercept term (when the driver variables are set at one) is far away from the mass of the data points. In this circumstance, the variance in the estimate of the intercept can be very large, larger in fact than the population variance of the regression line. Using the median value of Y at the intercept (e.g., the first unit cost in learning curves) to correct the upward bias of the median for the entire equation in unit space can cause the corrected function to lie outside the range of the data set. This practice should be avoided because Goldberger's Factor should be evaluated point by point.

The primary use of a CER is to make future predictions based on future driver values, which may or may not be in the CER data range. However, the use of a CER for extrapolation is always risky, especially for log-error models. Given the wide availability of computers, analysts may use Goldberger's equation for prediction even if the future x value lies outside the data range. But if the future prediction is far away from the center of the database, ***Goldberger's equation will not produce a logical or intuitively correct result*** as illustrated above, especially when the standard error of estimate is moderately large. This pitfall is a big concern when using log-linear CERs.

Generally, the PING Factor should be applied to all equations fit in log space by re-specifying each equation's constant term as the product of its original constant and the correction factor. However, exercise caution if dummy variables are used to stratify observations with different attributes (e.g., airborne versus ground-based antennas). A CER's predictive capability may not be improved by applying one adjustment factor to two or more different populations if the individual sample variances associated with these different categorical data are not equal. In theory, these different populations should have similar variances (and similar slope parameters) in order to be analyzed in one equation with dummy variables just to differentiate the intercept term. However, for small samples, it is often hard to find sufficient evidence to reject the null hypothesis that the variances are equal. A more detailed discussion of the PING Factor and the derivation of Equation 34 can be found in Reference 1.

CONCLUSIONS

The log-error model and MUPE model are two popular techniques used to hypothesize the multiplicative error term in the CERs. If the multiplicative error term follows a log-normal distribution in unit space, then the use of log-error model is appropriate. If the multiplicative error term is symmetrical around one with a mean of one and variance of σ^2, then choose the MUPE method. Therefore, the choices between the MUPE and log-error models should be based upon the error term assumption.

There are pros and cons associated with different fitting techniques. The advantages of using the log-error model are given below:

- If the hypothesized equation is log-linear, e.g., $y = ax^b\varepsilon$, then the regression can be done in log space linearly under the logarithmic transformation. As a result, this process is an OLS

in log space and all the goodness-of-fit measures can be evaluated in that space. This advantage does not exist if the CER has a non-linear functional form in unit space but it cannot be transformed to a linear equation in log space.

- The standard error of estimate in log space (SEE_L) can be regarded as a measure of a percentage error at a certain given x level in unit space, i.e., $SEE_L \cong CoV_A$ at a given x level. Note that CoV_A denotes the coefficient of variation in the unit space expressed as a percentage. See Reference (1) for details.
- Log-errors (Equation 2) can be viewed approximately as the MUPE percentage error.

The disadvantages of using the log-error model are summarized below:

- It involves a two-step process. First, we need to transform both the dependent and independent variables to perform OLS in log space. After developing the CER in log space, we need to transform the results back to unit space.
- We need to derive a correction factor (by either Goldberger's method or the PING Factor) to adjust the unit space CER result to obtain an unbiased estimate, since the CER result is closer to the median than the mean in unit space. This is yet another extra step.
- We must be extremely cautious when the future prediction lies outside the data range. **Goldberger's Factor is not recommended when the driver variables are outside the range of the data used to create the CER**.
- The PING Factor is easier to use, just as accurate as the Goldberg Factor within the data range, and far more suitable outside the data range (although care must be taken).

Here are some salient points when using Goldberger's Factor or the PING Factor to achieve the unbiased estimate in unit space for log-linear CERs:

- Goldberger's Factor (Equation 36) and the PING Factor generally match each other very closely within the data range. There are two terms involved in both factors: one is for adjusting the downward bias between the mean and the median (a transformation bias); the other is used to adjust the upward bias for estimating the median (a sampling bias).
- Goldberger's Factor is a **variable** factor. It should be evaluated **point by point** and multiplied to the log-error CERs for the entire function in order to obtain the theoretical mean in unit space. This process is tedious, as shown in Equations 35 and 36; it can become very cumbersome when more independent variables are introduced into the equation.
- The PING Factor is a handy, **constant** factor, which is used to adjust the level of the **entire** function.
- A common misuse of Goldberger's Factor is to derive an adjustment at some extreme point, such as T1, and then multiply it to the entire equation. This practice employs the corrected equation well outside the majority of the data points and should be avoided.
- The PING Factor and MUPE equations match each other closely in most cases (Reference 3).
- The PING Factor should be used with caution if dummy variables are specified in the equation. This is because a constant correction factor (i.e., the PING Factor) may not be adequate to correct the downward bias for two or more populations with "possible" unequal variances.
- When making a prediction outside the data range, the theoretical unbiased Goldberger Factor should be used with caution because this factor may be considerably less than one when the prediction lies outside the data range.

Biography

Dr. Shu-Ping Hu: Educated at National Taiwan University (B.S., Mathematics) and University of California, Santa Barbara (M.S., Mathematics and Ph.D., Statistics). Dr. Hu is a technical expert at Tecolote Research, Inc. She joined Tecolote in 1984 and has served as a company expert in all statistical matters. She is experienced in independent cost estimation, cost research, and risk analysis. She advocated an iterative regression technique (MUPE) to model a multiplicative error term without bias and developed correction factors (the PING Factor) to adjust the downward bias in log-error models. She has over 10 years of experience in USCM CER development and the related database. She also has 17 years of experience in the design, development, modification, and integration of statistical software packages for fitting various types of regression equations, learning curves, cost risk analysis, and other PC-based models.

Tel: (805) 964-6963
Fax: (805) 964-7329
E-mail: shu@tecolote.com

REFERENCES

1. Hu, S. and Sjovold, A. R., "Error Corrections for Unbiased Log-Linear Least Square Estimates," TR-006/2, March 1989.

2. Hu, S. and Sjovold, A. R., "Multiplicative Error Regression Techniques," 62nd MORS Symposium, Colorado Springs, Colorado, 7-9 June 1994.

3. Hu, S., "The Minimum-Unbiased-Percentage-Error (MUPE) Method in CER Development," 3rd Joint Annual ISPA/SCEA International Conference, Vienna, VA, 12-15 June 2001.

4. Goldberger, A. S., "The Interpretation and Estimation of Cobb-Douglas Functions," Econometrica, Vol. 35, July-Oct 1968, pp. 464-472.

5. Nguyen, P., N. Lozzi, et al., "Unmanned Space Vehicle Cost Model, Eighth Edition," U. S. Air Force Space and Missile Systems Center (SMC/FMC), October 2001.

6. Hillebrandt, P., Killingsworth, P., et al., "Unmanned Space Vehicle Cost Model, Sixth Edition," U. S. Air Force Space Division (AFSC), November 1988.

7. Duan, N., "Smearing Estimate: A Nonparametric Retransformation Method," Journal of the American Statistical Association, Vol. 78, Sep 1983, No. 383, pp. 605-610.

8. Seber, G. A. F., and C. J. Wild, "Nonlinear Regression," New York: John Wiley & Sons, 1989, pages 37, 46, 86-88.

9. Weisberg, S., Applied Linear Regression, 2nd Edition," New York: John Wiley & Sons, 1985, pages 87-88.