Paper 263-25

Improved Confidence Intervals on the Mean Response in Simple Linearizable Regression

Mark Carpenter and Pandu Kulkarni, University of South Alabama, Mobile, Alabama

ABSTRACT

Suppose that a response variable and its predictor variable are nonlinearly related but their relationship is linearizable through transformation, such as the log or square root transform on the response variable. In this situation, confidence intervals on the mean response at a given level of the predictor variable are traditionally done by first calculating the confidence bounds on the mean of the transformed response variable and then taking the inverse transform of these bounds. Although accurate for predication intervals, this ad hoc approach typically yields extremely biased intervals that do not produce the intended level of confidence for the mean response. In this paper, we propose an alternative approach using bootstrapped percentile intervals and bootstrap bias corrected intervals. This procedure will involve repeated application of the MACRO "jackboot.sas", provided by the SAS® institute, PROC REG, and DATA STEP programming. In addition to the theoretical development in this paper, we include a simulation study in which SAS is used to demonstrate that the bootstrapped intervals produce more accurate coverage. Also, we provide detailed instructions on how to use the "jackboot.sas" macro in these situations. The audience need only be familiar with simple linear regression and have a minimal exposure to SAS MACRO programming.

INTRODUCTION

In this section we introduce the simple linearizable regression models. To do so, we first review the *simple linear regression* model between a dependent (response) variable y and an independent (predictor) variable x, which is typically expressed as

$$y = \alpha + \beta x + \varepsilon \tag{1}$$

where ε, the error term, is a random variable with mean 0 and variance σ^2.

Under the construct given in (1) and using *n* sample data points, (x_1, y_1), (x_2, y_2),…, (x_n, y_n), it is easy to derive the least square sample estimates of α and β, say $\hat{\alpha}$ and $\hat{\beta}$. However, very often, researchers are more interested in accurately predicting a new value of y or estimating the mean of *y* at a given level of *x*. Using $\hat{\alpha}$ and $\hat{\beta}$, a point estimate y at a given level of x is given as

$$\hat{y} = \hat{\alpha} + \hat{\beta} x. \tag{2}$$

If the error term is normally distributed, then it is well known that $\hat{y}$ is normally distributed with mean and variance,

$$\mu_{\hat{y}} = E(\hat{y} \mid x) = \alpha + \beta x \text{ and } \sigma^2_{\hat{y}} = \sigma^2 \left[\frac{1}{n} + \frac{(x - \bar{x})^2}{s_{xx}}, \right] \tag{3}$$

where $s_{xx} = \sum (x_i - \bar{x})^2$. Accordingly, the (1-α) 100% confidence intervals on the mean and the prediction intervals, denoted as L^c_y / U^c_y and L^p_y / U^p_y respectively, are

Confidence: $$L^c_y / U^c_y = \hat{y} \pm t_{\alpha/2, n-2} \sqrt{MSE} \sqrt{\left[\frac{1}{n} + \frac{(x - \bar{x})^2}{s_{xx}} \right]} \tag{4}$$

Prediction: $$L^p_y / U^p_y = \hat{y} \pm t_{\alpha/2, n-2} \sqrt{MSE} \sqrt{\left[1 + \frac{1}{n} + \frac{(x - \bar{x})^2}{s_{xx}} \right]} \tag{5}$$

where $MSE = \frac{1}{(n-2)} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$.

Sometimes the relationship between two variables is not linear, i.e., the relationship is different than (1), but often the relationship is *intrinsically linear*. We say that the regression relationship between a dependent variable z and an independent variable x is intrinsically linear if there exists a monotone function (such as the natural log or the square-root function) $g(\cdot)$ such that

$$y = g(z) = \alpha + \beta x + \varepsilon. \tag{6}$$

In this situation, if one wishes to predict a future value of z or estimate the mean of z at a particular value of x, it is standard practice to take the inverse transform $g^{-1}(\cdot)$ of the predicted value or the estimated mean value of y given in (2),

$$\hat{z} = g^{-1}(\hat{y}) = g^{-1}(\hat{\alpha} + \hat{\beta} x). \tag{7}$$

Also, if it can be assumed that the error term in (6) is normally distributed with mean 0 and variance σ^2, you can similarly get confidence bounds on the point estimate given in (7), by taking the inverse transform of the bounds given in (4) and (5),

Confidence: $$\left[L^c_z, U^c_z \right] = \left[g^{-1}\left(L^c_y \right), g^{-1}\left(U^c_y \right) \right] \tag{8}$$

Prediction: $$\left[L^p_z, U^p_z \right] = \left[g^{-1}\left(L^p_y \right), g^{-1}\left(U^p_y \right) \right] \tag{9}$$

In the next section, we show that the prediction intervals in (9) give the intended coverage probability while the confidence intervals in (8) do not. In fact, we will show mathematically and through simulation that the actual coverage probabilities tend to be much smaller than expected. Example 1, given below, is based on a data set in which there is strong evidence that the relationship between the variables is intrinsically linear through the natural log transformation on the dependent variable.

Example 1: Devore (1995, page 535) introduced a data set from *Plant Physiology* in which the ethylene content of lettuce seeds (z, in Nl/g dry wt) is strongly log-linearly related to exposure time to an ethylene absorbent (x, in minutes). The SAS code in Figure 1 was used to generate the data in Table 1. The dependent variable y is calculated by taking the natural log of the values of z. Simple linear regression is performed between y and x and the 95% confidence intervals on the mean of y for each value of x is calculated. Finally, the bounds on z are found by the exponentiation of the bounds on y. The fitted regression line on y is given in (10) and the corresponding estimates of z are given in (11) below

$$\hat{y} = 5.941 - 0.032x \tag{10}$$

$$\hat{z} = \exp(\hat{y}) = \exp(5.941 - 0.032x). \tag{11}$$

We will not go into the details as to the appropriateness of the fitted equation in (10), but as Devore points out, the fitted equation seems to give an extremely good fit to the sample data, e.g., $\sqrt{\text{MSE}} = 0.078$ and R^2=0.9952.

If the intervals, based on (8) and given in Table 1, on the mean ethylene content of lettuce at each level of x are accurate then the standard interpretation would be appropriate. For example, we are 95% sure that the true mean ethylene content of lettuce seeds (z) after 30 minutes of exposure (x) to an ethylene

absorbent is between 135.28 and 153.53 Nl/g. However, as we shall see in the remainder of this section and subsequent sections, the actual coverage probabilities may be much less than reported.

Table 1: Example 1 Data and Intervals

x	z	$y=\ln z$	$\hat{y}$	$\hat{z}=e^{\hat{y}}$	L_y^c	U_y^c	L_z^c	U_z^c
2	408	6.01	5.88	356.32	5.78	5.97	323.21	392.83
10	274	5.61	5.62	275.12	5.53	5.70	252.33	299.97
20	196	5.28	5.29	199.12	5.22	5.37	184.95	214.38
30	137	4.92	4.97	144.12	4.91	5.03	135.28	153.53
40	90	4.50	4.65	104.31	4.59	4.70	98.64	110.31
50	78	4.36	4.32	75.49	4.27	4.38	71.58	79.62
60	51	3.93	4.00	54.64	3.95	4.06	51.68	57.77
70	40	3.69	3.68	39.55	3.62	3.74	37.14	42.12
80	30	3.40	3.35	28.62	3.28	3.43	26.60	30.80
90	22	3.09	3.03	20.72	2.95	3.12	19.01	22.58
100	15	2.71	2.71	14.99	2.61	2.81	13.57	16.57

Figure 1: SAS Program to Generate Data for Exmple 1 given in Table 1.

```
DATA DATA1;
  INPUT x z @@;
   y=log(z);
CARDS;
2 408 10 274 20 196 30 137 40 90 50 78
60 51 70 40 80 30 90 22 100 15
;
PROC REG DATA=data1 NOPRINT;
 MODEL y = x;
  OUTPUT OUT=data1 P=pred R=resid
                    L95M=ALCL_y U95M=AUCL_y;
RUN;
DATA data1;
  SET data1;
  ALCL_z=exp(ALCL_y);
  AUCL_z=exp(AUCL_y);
  Zhat=exp(pred);
RUN;

PROC PRINT DATA=data1 NOOBS ROUND;
 VAR x z y Zhat pred ALCL_y AUCL_y ALCL_z AUCL_z;
RUN;
```

Inspired by Example 1, in Tables 2a through 2d, we explore the properties of coverage probability through several simulation studies. For each case, we generated 1000 confidence intervals for all 11 unique values of x given in the actual data set. In these simulations, the exact relationship between x and y is known and assumed to be $y = 5.941 - 0.032x + \varepsilon$ and the errors are normally distributed with mean 0 and standard deviations 0.08, 0.5, 1, 2, and 3, respectively. You can see that the empirical coverages on the mean are very close to 0.95 when the variance is 0.08 (very small). In Table 2b you see that the coverage is less that 0.95. The rest of the tables show us that as the variance gets larger, the actual coverage probabilities decrease. In fact, in Table 2e we see that the empirical coverage is very much less than the intended 0.95. In short, for the log-linearizable situation, the coverage probabilities decrease dramatically as the correlation between x and y goes down and that only as the variance goes to zero (or the correlation goes to 1) do we achieve the intended coverage.

During the course of our research, many other simulation studies were conducted, such as the ones based on Example 1. However, for the sake of brevity, we limit this paper to just a few examples to make our case and state that throughout each simulation we have found similar patterns. Moreover, we mathematically prove in the next section that the actual confidence using the inverse transform method is not equal to the stated coverage.

Table 2a: Simulation Based on Example 1 with $\sigma = 0.08$

	Mean Confidence Limits		Empirical Coverage
μ_z	ALCL	AUCL	Probability
357.49	324.44	394.13	0.96
276.08	253.15	300.81	0.95
199.88	185.43	214.86	0.96
144.71	135.56	153.80	0.96
104.76	98.79	110.45	0.95
75.85	71.67	79.69	0.95
54.91	51.72	57.81	0.95
39.75	37.16	42.13	0.95
28.78	26.61	30.81	0.96
20.84	19.02	22.58	0.96
15.08	13.57	16.57	0.96

Table 2b: Simulation Based On Example 1 with $\sigma = 0.5$

	Mean Confidence Limits		Empirical Coverage
μ_z	ALCL	AUCL	Probability
403.79	205.66	689.95	0.93
311.84	168.10	491.63	0.92
225.77	130.04	325.15	0.91
163.45	99.62	218.48	0.91
118.33	75.04	150.25	0.90
85.67	55.09	106.65	0.90
62.02	39.27	78.47	0.91
44.90	27.27	59.61	0.91
32.51	18.62	46.37	0.92
23.54	12.58	36.66	0.92
17.04	8.46	29.31	0.92

Table 2c: Simulation Based On Example 1 with $\sigma = 1$

	Mean Confidence Limits		Empirical Coverage
μ_z	ALCL	AUCL	Probability
587.51	129.05	1480.70	0.88
453.73	109.93	955.15	0.86
328.49	89.51	566.48	0.81
237.82	71.83	348.80	0.76
172.17	55.97	226.24	0.70
124.65	41.65	157.22	0.67
90.24	29.33	118.05	0.71
65.33	19.72	95.02	0.76
47.30	12.87	80.62	0.81
34.24	8.27	71.05	0.85
24.79	5.29	64.40	0.88

Table 2d: Simulation Based On Example 1 with $\sigma = 2$

	Mean Confidence Limits		Empirical Coverage
μ_z	ALCL	AUCL	Probability
2633.05	67.45	9056.91	0.62
2033.47	58.81	4455.07	0.55
1472.18	49.86	1977.27	0.42
1065.82	41.90	970.92	0.30
771.63	33.96	543.02	0.18
558.64	25.72	357.65	0.14
404.44	17.91	282.09	0.19
292.80	11.67	262.40	0.28
211.98	7.34	278.75	0.43
153.47	4.60	328.94	0.55
111.11	2.92	423.90	0.65

Table 2e: Simulation Based On Example 1 with $\sigma = 3$

μ_z	Mean Confidence Limits ALCL	AUCL	Empirical Coverage Probability
32077.18	48.50	87591.71	0.33
24772.76	40.69	30626.49	0.22
17934.83	33.83	9364.71	0.11
12984.35	28.52	3418.68	0.04
9400.33	23.40	1563.67	0.02
6805.59	17.81	943.56	0.01
4927.07	12.32	769.61	0.01
3567.07	7.95	827.94	0.04
2582.47	5.04	1118.56	0.11
1869.64	3.28	1820.33	0.21
1353.57	2.23	3473.20	0.34

2. COVERAGE PROBABILITY

In this section, we show that the stated coverage is accurate for the prediction intervals using the inverse transform method, but the same is not true for confidence intervals on the mean response. We have already seen some indication of this through simulation studies in the previous section where in the log-linearizable case the actual coverage probability tended to be much less than intended.

We begin with the prediction interval, given in (9), for a new value of z (through y) at a particular value of x. Based on the normality properties we know, from (5), that for the (1-α)100% prediction interval

$$P\left[L_y^p < y < U_y^p\right] = 1 - \alpha.$$

Taking the inverse transform for each term within the brackets, we see that

$$P\left[g^{-1}\left(L_y^p\right) < g^{-1}(y) < g^{-1}\left(U_y^p\right)\right] = 1 - \alpha.$$

Therefore, since by definition of $z = g^{-1}(y)$, we have

$$P\left[g^{-1}\left(L_y^p\right) < z < g^{-1}\left(U_y^p\right)\right] = 1 - \alpha.$$

That is, the coverage probability for the (1- α) prediction intervals given in (9) on the original dependent variable *z* give the intended coverage probability of (1- α).

We now show that the confidence intervals on the mean given in (8) do not necessarily give the intended coverage. From (4) we have that

$$P\left[L_y^c < \mu_y < U_y^c\right] = 1 - \alpha.$$

This implies that

$$P\left[g^{-1}\left(L_y^c\right) < g^{-1}(\mu_y) < g^{-1}\left(U_y^c\right)\right] = 1 - \alpha.$$

However, it is a well known fact that for any function, other than the identity function,

$$\mu_z = E(z \mid x) = Eg^{-1}\left(\mu_y\right) \neq g^{-1}(\mathrm{E}(y \mid x)) = g^{-1}(\mu_y).$$

Therefore,

$$P\left[g^{-1}\left(L_y^c\right) < g^{-1}(\mu_y) < g^{-1}\left(U_y^c\right)\right] \neq P\left[g^{-1}\left(L_y^c\right) < \mu_z < g^{-1}\left(U_y^c\right)\right]$$

From the above relation, we see that the stated coverage for the mean of *z* is seriously in question. In fact, as we saw in the simulation study in the previous section, we shall show that in many common cases, the inverse transform method in (8) gives far less coverage than supposed

Example 2: Suppose the relationship between *z* and *x* is log-linearizable, i.e., the appropriate transformation is the natural log-transform $g(z) = \ln z$, then

$$\mu_z = E(z \mid x) = E\exp\{y\} \neq \exp\{\mu_y\}. = \exp\{E(y)\}.$$

Moreover, if we assume that the errors in (6) are normal with mean 0 and variance σ^2 , then *z* is a log-normal random variable with mean and variance

$$\mu_z = \exp\{\mu_y + \sigma^2/2\} \quad \text{and} \quad Var(z) = \exp\{2\mu_y + \sigma^2/2\}. \qquad (10)$$

Thus, from (10) we see that statistics based on the inverse transform can be extremely biased and the variance may be extremely underestimated as the variance in *y*, σ^2 , increases. In fact, it can be shown that

$$E\exp\{\hat{y}\} = E(z) \cdot \exp\left\{\frac{\sigma^2}{2}\left(\frac{1}{n} + \frac{(x-\bar{x})^2}{s_{xx}} - 1\right)\right\}. \qquad (11)$$

The second term on the right-hand-side of equation (11) is always either less than or greater than 1 and, therefore, the bias increases as σ^2 increases. Also, we see that this term approaches 1 as σ^2 goes to zero (or as the correlation approaches 1). Recall that in Table 2a the empirical coverages were close to the stated coverage.

One way to improve upon our estimation technique is to use the MLE or other plug-in estimators of each of the unknown parameters. That is, we plug in estimators of μ_y and σ^2 into the mean response given in (10) as follows,

$$\begin{aligned} \hat{\mu}_z &= \exp\{\hat{\mu}_y + \hat{\sigma}^2/2\} = \exp\left\{\hat{y} + MSE/2\right\} \\ &= \exp\left\{\hat{\alpha} + \hat{\beta}x + MSE/2\right\} \end{aligned} \qquad (12)$$

Similar plug-in estimators can be derived for the other transforms as well, such as the square-root transform or one of the many Box-Cox type power transforms.

The Box-Cox class of transformations is the large class of power transformations given as

$$g(z) = z^{\lambda},$$

where λ is either given priori or is estimated from the data. Note that the above class includes, among others, the log and the square-root transforms. From Neter et al. (1996), we see that

$$\begin{aligned} &\lambda = 2; && g(z) = z^2 \\ &\lambda = 0.5; && g(z) = \sqrt{z} \\ &\lambda = 0; && g(z) = \ln(z) \\ &\lambda = -0.5; && g(z) = \frac{1}{\sqrt{z}} \\ &\lambda = -1.0; && g(z) = \frac{1}{z}. \end{aligned} \qquad (13)$$

The Box-Cox linearizable model is as follows

$$y = g(z) = \beta_0 + \beta_1 x + \varepsilon.$$

Assuming that the errors are normally distributed with mean 0 and standard deviation σ^2, the Box-Cox procedure finds the Maximum Likelihood Estimates of $\beta_0, \beta_1, \sigma^2$ and λ and the MLE for the mean value of y at a particular level of x is given as

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x.$$

and a naive point estimate for a predicted value of z (or mean value of z) is

$$\hat{z} = g^{-1}(\hat{y}) = \begin{cases} (\hat{\beta}_0 + \hat{\beta}_1 x)^{1/\hat{\lambda}}, & \text{if } \hat{\lambda} \neq 0 \\ \exp(\hat{\beta}_0 + \hat{\beta}_1 x), & \text{if } \hat{\lambda} = 0. \end{cases} \quad (14)$$

In this paper, we will not delve into a large number of the Box-Cox transformations because to get reasonable plug-in estimators of z you must derive its expected value. For each of the transforms given in (13), these derivations are fairly straightforward. For the purposes of illustration, we continue with the theme of the inverse log transform.

3. BOOTSTRAP CONFIDENCE INTERVAL

In Example 1 the values of the predictor variable are fixed (not random); throughout this paper we will assume the fixed design. Accordingly, to bootstrap any statistic derived from simple linear regression, such as the correlation coefficient, instead of getting the bootstrap samples from the n pairs of data, we must sample from the residuals created from the fitted regression. The basic method for generating B bootstrap observations for both y and z is given in Figure 2. As mentioned in the previous section, we must develop a reasonable plug-in estimate of z in order for the bootstrap scheme to give reasonable results. In each of the situations that we explored the naïve estimates of z given in (14) gave no improvement over the intervals given in (8). We had hoped to develop a nonparametric approach whereby the bootstrap confidence intervals would be calculated from (14) instead of having to derive expected values and deriving plug-in estimates such as (12). Nevertheless, our approach yields remarkable improvement over the traditional approach (8).

Figure 2: Algorithm for the Linearizable Regression Bootstrapped Confidence Intervals on a Mean.

- **Step 1:** Using the *n* sample points, calculate the least squares fitted values $\hat{y} = \hat{\alpha} + \hat{\beta}x$ given in (2); See the PROC REG given in Figure 1.
- **Step 2:** Calculate the n sample errors (residuals) for the fits in Step 1, $e_i = (y_i - \hat{y}_i)$.
- **Step 3:** By sampling with replacement from $e_1, e_2, \ldots, e_n$ with equal weight on each observation, collect the bootstrap sample errors $e_1^*, e_2^*, \ldots, e_n^*$.
- **Step 4:** The bootstrap responses for the transformed variable (*y*) are then generated by $y_i^* = \hat{\alpha} + \hat{\beta}x_i + e_i^*$.
- **Step 5:** The bootstrap responses for the response variable (z) are then generated by the plug-in estimates, z_i^*, using y_i^* from Step 4; See the log-transform plug-in estimator given in (12).
- **Step 6**: Repeat Steps 3-5 a total of B times, so that for each x_i a total of B bootstrap observations are generated.

The resulting B bootstrap observations represent the bootstrap distribution of $\hat{z}$. The bootstrap confidence intervals can be calculated by several methods using the "jackboot.sas" macro. In this paper the focus is on the bootstrap percentile method and the bootstrap bias corrected method. For detailed information about these methods you can see the full documentation contained in the "jackboot.sas" file (see Sarle (1997)) or Efron and Tibshriani (1993). As Sarle describes, the percentile bootstrapping method calculates the $\alpha/2$ and 1-$\alpha/2$ percentiles of the bootstrap distribution of the statistic of interest (plug-in estimator). Typically these statistics are biased and, as Sarle points out, the percentile method will increase the bias (and as discussed in the previous section, our statistics are biased). The Bootstrap Bias Corrected method (BC) corrects the percentile interval for median-bias. The correction is performed by adjusting the percentile points to values other than $\alpha/2$ and 1-$\alpha/2$ percentiles. For more details of these methods the reader is referred to Efron and Tibshirani (1993) and Hall (1992).

In the remainder of this section, we evaluate and compare and contrast the bootstrap intervals to the traditional inverse-transform method in (8). In Section 4, we show you how to actually use the "jackboot.sas" macro to get these improved intervals.

Figure 2 gives the empirical coverage probabilities based on 1000 simulated intervals. For the bootstrap distributions we chose to use B=200. The regression is based on 20 observed pairs of data (x,y), with the linearized relationship $y = 10 + 4x + \varepsilon$, where the errors are normally distributed with mean 0 and variance 3. The values of x were chosen by a normal randomization scheme with mean 8 and variance 4. We do not provide an extensive number of simulations, as the full simulations are extremely computationally expensive; we only provide an excerpt here. The results given in Figure 2 took several days to run on a fairly fast computer. Many other simulation studies were run with smaller numbers of intervals and much smaller bootstrap sample sizes. In every case, the bootstrap method gave substantial improvement, except in situations where the variance in y is extremely close to zero (in these cases the bootstrap intervals give equivalent results to the standard transform method). You can contact the authors if you wish to obtain the SAS code that was used.

As we can see from Figure 2, the bootstrap methods give remarkable improvement of the transform method.

Figure 2: Simulation Study Indicating Strong Improvement by Bootstrap, B=200.

Standard Transform	Bootstrap Percentile	Bootstrap Bias Corrected
0.55	0.87	0.88
0.13	0.84	0.89
0.05	0.84	0.90
0.04	0.84	0.85
0.04	0.84	0.86
0.02	0.84	0.86
0.02	0.83	0.86
0.01	0.84	0.82
0.01	0.84	0.82
0.01	0.84	0.80
0.00	0.84	0.81
0.01	0.83	0.82
0.02	0.83	0.81
0.02	0.84	0.83
0.14	0.87	0.83
0.15	0.87	0.86
0.25	0.85	0.86
0.34	0.85	0.87
0.46	0.85	0.87
0.52	0.86	0.88

4. USING THE JACKBOOT MACRO

In this section we demonstrate how to use the "jackboot.sas" MACRO to get the bootstrap confidence intervals on the mean response for a fixed value of the predictor variable. The basic algorithm for generating bootstrap responses is described by Steps 1-4 in Figure 2. That is, to get bootstrap estimates using simple linear regression when the values of the predictor variables are fixed, you must use the algorithm given in Figure 2 instead of sampling pairs of data. To do so the "jackboot" macro requires you to create an input data set containing the predicted values of y and the residuals resulting from the OUTPUT statement in PROC REG. Figure 3a contains the SAS code that creates the required input data set for Example 1.

Figure 3a: Required form of the Input Data Set for Bootstrapping Residuals.

```
DATA DATA1;
  INPUT x z @@;
    y=LOG(z);
CARDS;
2 408 10 274 20 196 30 137 40 90 50 78
60 51 70 40 80 30 90 22 100 15
;

PROC REG DATA=data1;
  MODEL y = x;
   OUTPUT OUT=data1 P=pred R=resid;
RUN;
```

The next step is to write a SAS macro called "analyze". This macro is used to calculate an instance of the statistic of interest. The output data set will contain only the statistics of interest. Figure 3b contains the code for calculating a point estimate of z at a particular value of x. In this case, we let x0=30. You need only change the value of x0 to obtain the confidence interval of your choice.

Figure 3b: Required MACRO called "analyze" that you must create. The output data set will contain the statistics that you wish to bootstrap. You need only change the value of x0 to obtain the confidence interval of your choice.

```
%macro analyze (data=,out=);

PROC REG  DATA=&data   OUTEST=&out
     (KEEP=intercep x  RMSE
     RENAME=(x=beta  _RMSE_=sig_hat) )
     NOPRINT;
  MODEL y = x;
RUN;

DATA &out;
  SET &out;
  x0=30;
  yhat=intercep + beta*x0;
  zhat=exp(yhat+sig_hat**2/2);
  KEEP zhat;
RUN;

%mend analyze;
```

Now, to get the bootstrap confidence intervals you must run code similar to that given in Figure 3c. For the code to work as is, you must have downloaded the "jackboot.sas" macro from the SAS website, at **www.sas.com**, and saved it in your local SAS directory. Also, you must have created and then saved the analyze macro into your local SAS directory. Note that if either file is not saved in your local SAS directory, then you must supply the exact path name within the single quotes of the include statement.

Figure 3c: The "jackboot.sas" macro used to obtain the Bootstrap intervals on the mean response.

```
%include 'jackboot.sas';
%include 'analyze.sas';

%boot (data=data1, residual=resid,
       equation=y=pred+resid,
       samples=500,alpha=.05,
       random=123, print=0, char=0);

%bootci (percentile,alpha=.05, print=0);

PROC PRINT DATA=BOOTCI;
 VAR value alcl aucl;
  TITLE 'Bootstrap Percentile Method';
RUN;

%bootci (bc, alpha=.05,print=0);

PROC PRINT DATA=BOOTCI;
 VAR value alcl aucl;
  TITLE 'Bootstrap Bias Correction Method';
RUN;
```

A more extensive tutorial on the "jackboot.sas" macro can be found at http://www.mathstat.usouthal.edu/~dmc/sugi25.

CONCLUSION

In this paper, we showed that the prediction intervals derived from the inverse transform method (9) do give the appropriate level of confidence and, therefore, they need no adjustment. However, the confidence intervals on the mean response do not give the stated level of confidence and, in fact, they often give ridiculous results in terms of actual coverage. In short, unless there is an extremely strong relationship between the transformed variable (y) and the predictor variable (x) (correlation greater than 0.99 and variance in y close to zero), the transform method is very unreliable.

Through simulation study, we showed that the bootstrap percentile and the bias corrected methods offer substantial improvement over the transform method and are viable techniques to be used for linearizable regression models. Although, we were not able to achieve exactly the correct coverage in all cases, the coverage tends to be quite reasonable.

We demonstrated that the "jackboot.sas" MACRO is an extremely useful way to generate bootstrap confidence intervals, as well as, other bootstrap statistics such as the correlation, coefficient of variation, etc. We recommend that the novice SAS programmer in need of bootstrap statistics use this macro, as it is particularly easy to use and does not require substantial knowledge of the bootstrapping techniques. In the last section of this paper, we gave a brief tutorial on how to use the macro to generate the bootstrap percentile and the bootstrap bias corrected intervals for the mean response at a particular level of the predictor variable.

REFERENCES

Devore, Jay L. (1995). *Probability and Statistics for Engineering and the Sciences, 4th Edition*, Duxbury Press.

Efron, Bradley and Tibshirani, Robert J. (1993*). An Introduction to the Bootstrap*, Chapman and Hall.

Hall, P. (1992). *The Bootstrap and Edgeworth Expansion*, New York: Springer-Verlag.

Neter, John, Kutner, M.H., Nachtsheim, C.J. and Wasserman, William (1996), *Applied Linear Regression Models*, Irwin Press.

"Ethylene Synthesis in Lettuce Seeds: Its Physiological Significance," *Plant Physiology*, 1972, pp. 719-722.

Sarle, Warren S. (1997). *Jackboot SAS Macro and Documentation*. www.sas.com.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.
Contact the author at:

Mark Carpenter
University of South Alabama
Department of Mathematics and Statistics
Mobile, Alabama, 36688
Work Phone: 334-461-1505
Fax: 334-460-7960
Email: dmc@mathstat.usouthal.edu
Web: http://www.mathstat.usouthal.edu/~dmc