A STUDY OF THE CALIBRATION-INVERSE PREDICTION PROBLEM IN A MIXED MODEL SETTING

by

CELESTE YANG

B.S., Kansas State University, 2005

A REPORT

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2008

Approved by;

Major Professor

Paul I. Nelson

ABSTRACT

The Calibration-Inverse Prediction Problem was investigated in a mixed model setting. Two methods were used to construct inverse prediction intervals. Method 1 ignores the random block effect in the mixed model and constructs the inverse prediction interval in the standard manner using quantiles from an F distribution. Method 2 uses a bootstrap to estimate quantiles of an approximate pivotal and then follows essentially the same procedure as in method 1.

A simulation study was carried out to compare how the intervals created by the two methods performed in terms of coverage rate and mean interval length. Results from our simulation study suggest that when the variance component of the block is large relative to the location variance component, the coverage rate of the intervals produced by the two methods differ significantly. Method 2 appears to yield intervals which have a slightly higher coverage rate and wider interval length then did method 1. Both methods yielded intervals with coverage rates below nominal for approximately 1/3 of the simulation settings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my major professor, Dr. Paul Nelson for all his patience and guidance throughout writing this report. This report is truly a joint effort between myself and my major professor and could not have been completed without Dr. Nelson's direction.

I would also like to thank my committee members Dr. Higgins and Dr. Neil. They helped edit this report and gave me advice on how to explain my results for future projects.

Further, to quote John Lennon from a Beatles song "I get by with a little help from my friends." This report could not have been completed without the help, and most of all the support of my friends Kendra Kubin, Raymond McCubrey, and Smriti Shrestha.

Finally, completing this project was the last, and most difficult step in obtaining my master's degree, I know that there will be other more complex projects of this nature that I will have to do, but I've figured out how to jump this hurdle, and even if the next hurdle is bigger or harder to jump over, I've got an idea how to handle it. That said, although this MS report is not the greatest thing I shall ever complete, perhaps figuring out how to handle a large project like this is a great thing, and so I dedicate this report to my dad.

Chapter 1- Introduction

1.1: Problem Statement

This report proposes and studies a solution to what is called the calibration or inverse prediction problem in a mixed model setting where experimental units are selected from blocks that are treated as random effects. This problem was motivated by a study currently being carried out at Fort Riley, Kansas. One of the objects of the study is to measure "bare ground" coverage in military maneuver plots. Among other variables, the researchers are interested in measuring the density of plant vegetation in these plots at different time periods (the blocks**).** Let X_{tj} be the density measurement taken on the ground at time *t* at location *j*, as identified by some coordinate system, for example, longitude and latitude. Let Y_{tj} be the estimated density measurement by satellite at time *t* and location *j*. Assume that the Y_{tj}'s can be easily obtained (less labor intensive than ground measurements). We are interested in solving the following calibration problem. We have data **D** = $\{(X_{tj},Y_{tj}),\ t=1, \ldots, K;\ \ j=1, \ldots, n_t\}$ obtained from the ground and the satellite. Additionally, we have a density measurement Y_{sk} independent of **D** made at location *k* at some 'future' time *s* obtained from the satellite. The objective here is to estimate the corresponding, unobserved X_{sk} based on the data and the newly observed Y_{sk}. In particular, we are interested in constructing what is called a $1-\alpha$ *inverse prediction set S* computed from data **D** and Y_{sk} so that P($X_{sk} \in S$) = 1 – α. We use a random 'time' effect to model the possible dependence among responses measured during the same time period. Assume that for all time periods *t* and all locations *j*, Y_{tj}, is linearly related to $X_{tj} = x_{tj}$ by the model:

$$\begin{aligned} Y_{tj} &= \beta_o + \beta_1 x_{tj} + e_{tj}, \\ e_{tj} &= \eta_t + \varepsilon_{tj} \end{aligned} \qquad \textbf{(1.1)}$$

The random time components $\{\eta_t\}$ are assumed to be independently normally distributed N(0, σ_η^2), independent of the location errors $\{\varepsilon_{tj}\}$, which are taken to be independent N(0, σ_ε^2). Ground data is obtained at Fort Riley over intervals of time spaced far apart. Accordingly the ground has become so altered as to make our assumption - that responses measured at different periods of time are independent - a reasonable one.
Further, assume that the error terms $\{e_{tj}\}$ are independent of the ground measurements $\{X_{tj}\}$, whose joint distribution is free of the parameters $\{\beta_0, \beta_1, \sigma_\eta^2, \sigma_\varepsilon^2\}$.
This last assumption, allows inference to be carried out conditional on the observed ground cover values $\{X_{tj}\} = \{x_{tj}\}$. Given that inference is carried out conditional on the observed x's, inverse prediction sets S are often called *confidence sets*. Following common practice, we will focus on the case where S is an interval.

Our assumptions lead to the following covariance structure:

$$Cov(Y_{tj}, Y_{sk}) = Cov(e_{tj}, e_{sk}) = \begin{cases} 0, & t \neq s \\ \sigma_\eta^2 + \sigma_\varepsilon^2, & t = s \text{ and } j = k \\ \sigma_\eta^2, & t = s \text{ and } j \neq k. \end{cases} \qquad \textbf{(1.2)}$$

This model can be expressed as a split-plot design where η_t is the whole plot error – i.e. the random error for the whole-plot experimental unit. Here, η_t is the error for t^{th} whole plot experimental unit and ε_{tj} is the error for the subplot experimental unit. In this mixed model setting we wish to obtain set estimates for X_{sk}.

1.2: Proposed Solution

Let $\{\hat{\beta}_0, \hat{\beta}_1\}$ denote the maximum likelihood estimators of the regression parameters β_0 and β_1. When $\sigma_\eta^2 = 0$, as described below, standard 'inverse prediction sets' for x_{sk} may be obtained by inverting the quantity

$$\tilde{T} = \frac{Y_{sk} - \hat{\beta}_0 - \hat{\beta}_1 x_{sk}}{\sqrt{\hat{V}ar\left(Y_{sk} - \hat{\beta}_0 - \hat{\beta}_1 x_{sk}\right)}} \quad \textbf{(1.3)}$$

viewed as a function of x_{sk} with **D** and Y_{sk} set equal to their observed values. Correcting for the bias in the maximum likelihood estimator of σ_ε^2, a scaled version of $\tilde{T}$, denoted T, has a *t*-distribution with n-2 degrees of freedom when $\sigma_\eta^2 = 0$, and

$$T = \sqrt{\frac{n}{n-2}} \frac{Y_{sk} - \hat{\beta}_0 - \hat{\beta}_1 x_{sk}}{\sqrt{\hat{V}ar\left(Y_{sk} - \hat{\beta}_0 - \hat{\beta}_1 x_{sk}\right)}} \quad \textbf{(1.4)}$$

T^2 has an *F* distribution with 1 degree of freedom in the numerator and n-2 degrees of freedom in the denominator, Graybill (1976). However, the exact distribution of T or T^2 has not been determined when $\sigma_\eta^2 > 0$ and cannot be simply simulated because x_{sk} is an unknown quantity. This report proposes and investigates two solutions to this problem, (i) ignore the block effect and use a t-distribution with n-2 degrees of freedom; (ii) use quantiles obtained from a bootstrap. As in the standard case, we will use a two stage procedure where the inverse prediction interval is constructed if and only if H_0: $\beta_1 = 0$ is rejected in favor of H_a: $\beta_1 \neq 0$ (Graybill 1976). Simulation will be used to evaluate and compare these two solutions based on coverage rate and interval length.

1.3: An Example

To illustrate what we propose in section 1.2, consider the following example. Suppose we have measurements taken from the ground and satellite for times, K=8, and $n_1 = n_2 = ... = n_8 = 6$ locations at each time. Using SAS and the following parameter settings: slope (β_1) = 8, time variance (σ_η) = 5, and location variance (σ_ε) = 0.05 we simulated y_{tj} according to the model described in equation (1.1). This data is presented in table 1.3.1 on the following page. Similarly, by means of equation (1.1) we generated a "new" observation, y_{sk}= -0.19902 corresponding to x_{sk} = 0.09207. Both x_{tj}'s and x_{sk} were generated from a U(0, 1) distribution.

Using the dataset above, the statistical software SAS 9.1 was used to fit a standard least squares line and to construct one at a time 0.95 prediction intervals for $Y_{sk}=y_{sk}$ given by,

$$\hat{y} \pm t_{(1-\alpha/2:n-1)} S_n \sqrt{1+\frac{1}{n}+\frac{(x_{sk}-\overline{x})^2}{S_{xx}}} \qquad \textbf{(1.5)}$$

where $\hat{y}=\tilde{\beta}_0+\tilde{\beta}_1 x$, $\tilde{\beta}_0$, $\tilde{\beta}_1$, are the least squares estimates of intercept and slope, n=48 observations, and

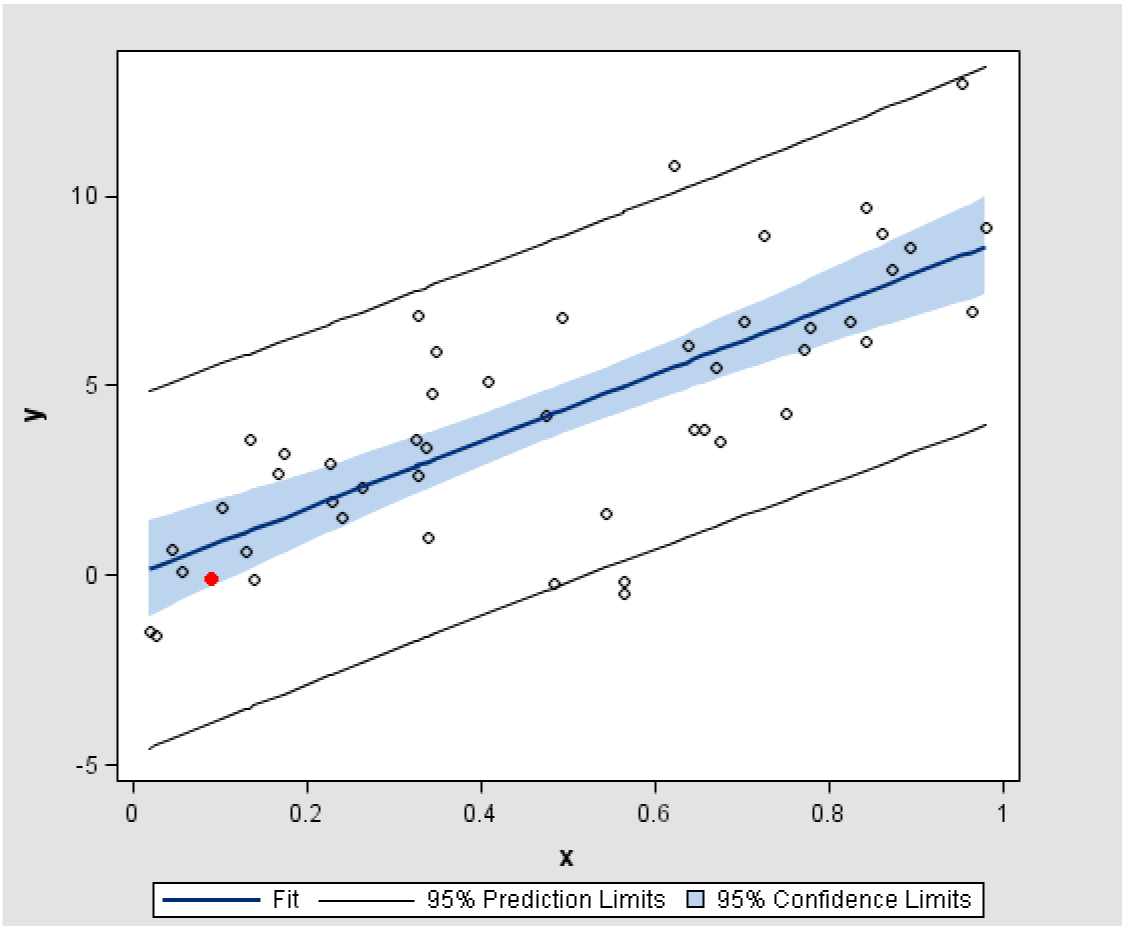
$$S_n=\sqrt{\frac{\sum_{i=1}^{n}(Y_i-\overline{Y})^2}{n-1}}, \qquad S_{xx}=\sum_{t=1,\dots,k,\, j=1,\dots,n_t}(x_{tj}-\overline{x})^2$$

A scatter plot of these standard least squares prediction intervals for y_{sk} is given in Figure 1.3.1. The target point $\mathbf{x_{sk}}$ **= 0.09207**, corresponding to $\mathbf{y_{sk}}$**= -0.19902**, appears as a red dot in the figure.

Table 1.3.1: Data $D = \{(X_{tj}, Y_{tj}),\ t=1, \ldots, 8;\ \ j=1, \ldots, n_t=6\}$

y_{tj}	x_{tj}
2.3414	0.26257
8.9471	0.72354
6.8076	0.49223
9.0282	0.85998
-0.1256	0.56269
6.525	0.77666
12.9706	0.95183
3.5447	0.67328
4.2364	0.47355
1.9865	0.22848
-0.1129	0.13894
3.5799	0.13257
2.9719	0.2258
4.2976	0.74852
8.6529	0.89218
3.8538	0.65574
5.9868	0.76902
3.6032	0.32458
-0.4854	0.56355
2.6332	0.32623
0.624	0.12862
-1.5436	0.02563
6.7194	0.70074
5.4871	0.6679
0.674	0.04329
9.7178	0.84142
3.3617	0.33593
6.8714	0.32671
-1.4588	0.0194
1.5276	0.23993
2.7107	0.16502
9.1589	0.98023
1.7803	0.10218
6.98	0.9628
3.8422	0.64433
6.0644	0.63606
6.2047	0.8407
0.113	0.05604
-0.2211	0.48219
6.7121	0.82239
3.2543	0.17164
5.1456	0.40807
5.9303	0.34821
1.6563	0.54267
4.8079	0.34307
1.0414	0.33842
8.0992	0.87013
10.8316	0.61968

Figure 1.3.1: A plot of the standard least squares prediction intervals for y_{sk}



As will be explained later, first ignoring the block effect, we used the 95th percentile of an F distribution with 1 df in the numerator and 46 df in the denominator to construct an approximate 95% inverse prediction interval for x_{sk} (method 1). Additionally, we bootstrapped the distribution of T^2 and used the 95th percentile of the bootstrapped distribution (F^*) to form an approximate 95% inverse prediction interval for x_{sk} (method 2) . The two intervals are given in table 1.3.2 below.

Table 1.3.2: Approximate 95% Inverse Prediction Intervals for x_{sk}

	95th percentile	lower bound	Upper bound
method 1	5.12197	-0.69963	0.5618
method 2	4.05175	-0.61373	0.4969

Note that both intervals contain the value of x_{sk}=0.092073 we are interested in predicting.

1.4: Organization of Remaining Chapters

In Chapter 2 we will show how we constructed approximate 1-α inverse prediction intervals for x_{sk} with the methods used in the example above. Chapter 3 outlines the simulation study that was used to compare the two methods. Chapter 4 explains the results of the simulation study, and Chapter 5 will summarize the findings of this report. Although many figures will be presented in the results chapter, for ease of reading it was necessary to place many of the tables and figures used to summarize our simulation results in appendices.

Chapter 2 – Inverse Prediction Interval

2.1: Background

The inverse prediction problem for a linear model with uncorrelated errors given by

$$Y_i = \beta_o + \beta_1 x_i + \varepsilon_{i,} \quad where\ \varepsilon_{i,} \sim N(0, \sigma_\varepsilon^2), \quad i = 1, 2, \ldots, n; \tag{2.1}$$

has been widely studied. See, for example, Brown (1979), Williams (1969), and Graybill (1976). Information on the robustness of these intervals may be found in Xiao (2000). However, to the best of my knowledge, the inverse prediction problem has not been studied in models with correlated errors. For the model with uncorrelated errors such as given in (2.1) above, Graybill (1976) developed a procedure for constructing an inverse prediction interval for a value x_o having observed $Y = y_o$.

2.2: Constructing an Inverse Prediction Set

We propose a method for constructing one-at-a-time interval estimates of the unobserved value x_{sk} corresponding to the observed value $Y_{sk} = y_{sk}$ in the mixed model setting presented in section 1.1. This method closely follows Graybill's procedure with a few adjustments. To illustrate, using equation (1.3), let

$$F = \tilde{T}^2 = \frac{(Y_{sk} - \hat{\beta}_0 - \hat{\beta}_1 x_{sk})^2}{\hat{V}ar\left(Y_{sk} - \hat{\beta}_0 - \hat{\beta}_1 x_{sk}\right)} \tag{2.2}$$

Assume that $\tilde{F}$ is pivotal so that the quantiles of its distribution, denoted $\left\{\tilde{F}_\gamma\right\}$, are free of unknown parameters and x_{sk}. Recall that when there is no block effect and the maximum likelihood estimators are corrected for bias, $\tilde{F}_{1-\alpha} = F_{1-\alpha:1,n-2}$, the $1-\alpha$ quantile from an F distribution with 1 degree of freedom in the numerator and n-2 degrees of freedom in the denominator, where $n = n_1 + n_2 + \ldots + n_t$. Then, replacing all the entries in (2.2) except x_{sk} by their observed values, a 1- α inverse prediction set, S for x_{sk} is given by

$$S = \left\{ x_{sk} : \tilde{F} \le \tilde{F}_{1-\alpha} \right\} \quad \textbf{(2.3)}$$

We proceed to solve for x_{sk} by first noting that, after some simplification, the inequality in (2.3) is equivalent to

$$\left(Y_{sk} - \hat{\beta}_0 - \hat{\beta}_1 x_{sk} \right)^2 - \hat{V}ar(Y_{sk} - \hat{\beta}_0 - \hat{\beta}_1 x_{sk})\tilde{F}_{1-\alpha} \le 0 \quad \textbf{(2.4)}$$

where,

$$\hat{V}ar(Y_{sk} - \hat{\beta}_0 - \hat{\beta}_1 x_{sk}) = \hat{\sigma}_\varepsilon + \hat{\sigma}_\eta + \hat{V}ar(\hat{\beta}_0) + x_{sk}^2 \hat{V}ar(\hat{\beta}_1) + 2x_{sk}\hat{C}ov(\hat{\beta}_0, \hat{\beta}_1)$$

Combining terms, we obtain a quadratic equation that is a function of x_{sk}

$$\begin{aligned} &\left[\hat{\beta}_1^2 - \hat{V}ar(\hat{\beta}_1)\tilde{F}_{1-\alpha} \right] x_{sk}^2 + 2\left[\hat{\beta}_o\hat{\beta}_1 - \hat{\beta}_1 Y_{sk} - 2\hat{C}ov(\hat{\beta}_o, \hat{\beta}_1)\tilde{F}_{1-\alpha} \right] x_{sk} \\ &+ \left[Y_{sk}^2 - 2\hat{\beta}_o Y_{sk} - \tilde{F}_{1-\alpha}(\hat{\sigma}_\varepsilon + \hat{\sigma}_\eta + \hat{V}ar(\hat{\beta}_1) \right] \le 0 \end{aligned} \quad \textbf{(2.5)}$$

which can be expressed as $q(x_{sk}) = ax_{sk}^2 + 2bx_{sk} + c \le 0$ where *a, b*, and *c* are straightforwardly obtained from (2.5) and given by

$$\begin{aligned} a &= \hat{\beta}_1{}^2 - \hat{V}ar(\hat{\beta}_1)\tilde{F}_{1-\alpha} \\ b &= \hat{\beta}_0\hat{\beta}_1 - \hat{\beta}_1 Y_{sk} - 2\hat{C}ov(\hat{\beta}_0, \hat{\beta}_1)\tilde{F}_{1-\alpha} \\ c &= Y_{sk}^2 - 2\hat{\beta}_0 Y_{sk} - \tilde{F}_{1-\alpha}(\hat{\sigma}_\varepsilon + \hat{\sigma}_\eta + \hat{V}ar(\hat{\beta}_1) \end{aligned} \quad \textbf{(2.6)}$$

The solutions to the quadratic equation obtained by setting the left hand side of (2.5) equal to zero can be one of the following:

Case 1: $a<0$ and $b^2-ac<0$: The resulting confidence interval is the whole real line.

Case 2: $a<0$ and $b^2-ac<0$: $b^2-ac>0$: the resulting confidence interval is the union of two semi-infinite pieces.

Case3: $a>0$ and $b^2-ac<0$: the confidence interval does not exist.

Case4: $a>0$ and $b^2-ac>0$: a confidence interval of finite length can be obtained.

We illustrate these cases with the subsequent figures:

Figure 2.2.1: Case 1: $a<0$ and $b^2-ac<0$

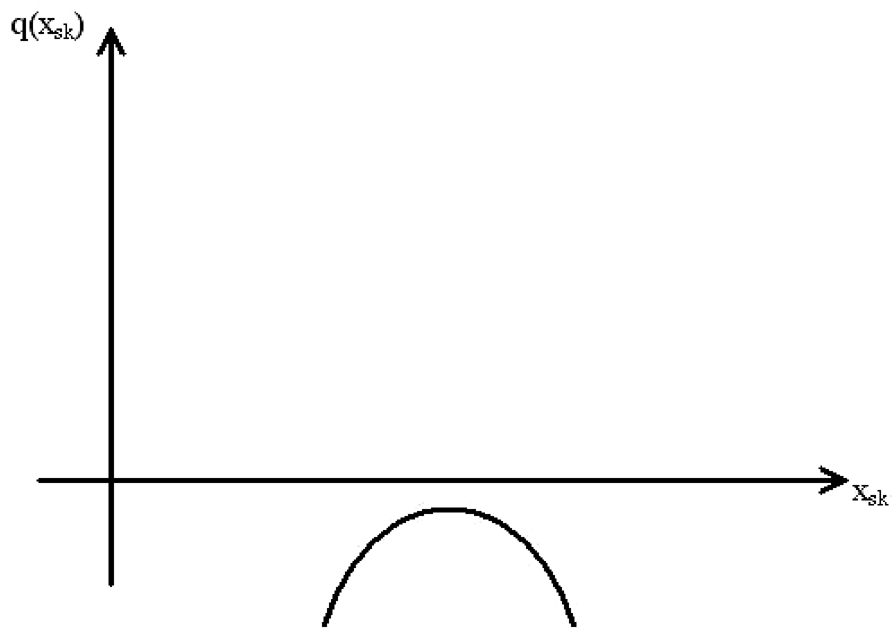


Figure 2.2.2: Case 2: a<0, b²-ac>0

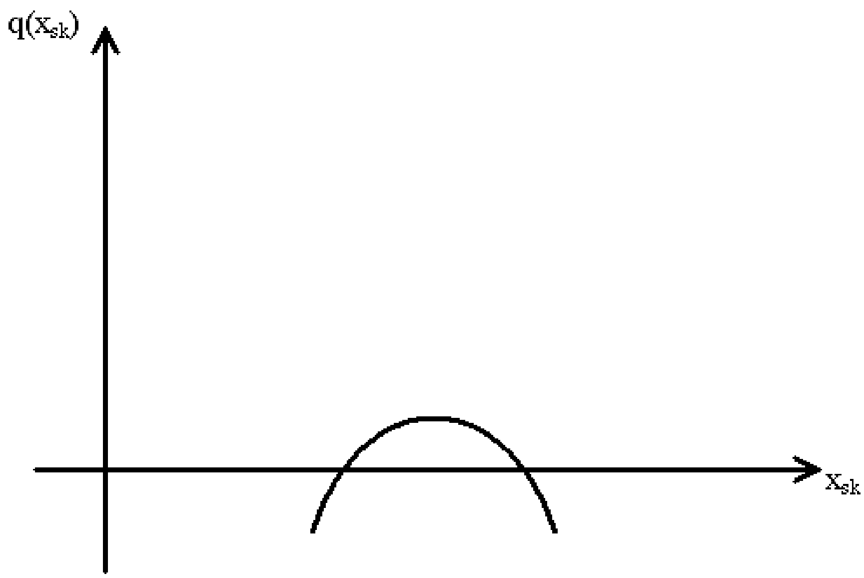


Figure 2.2.3: Case 3: a>0 and b²-ac<0

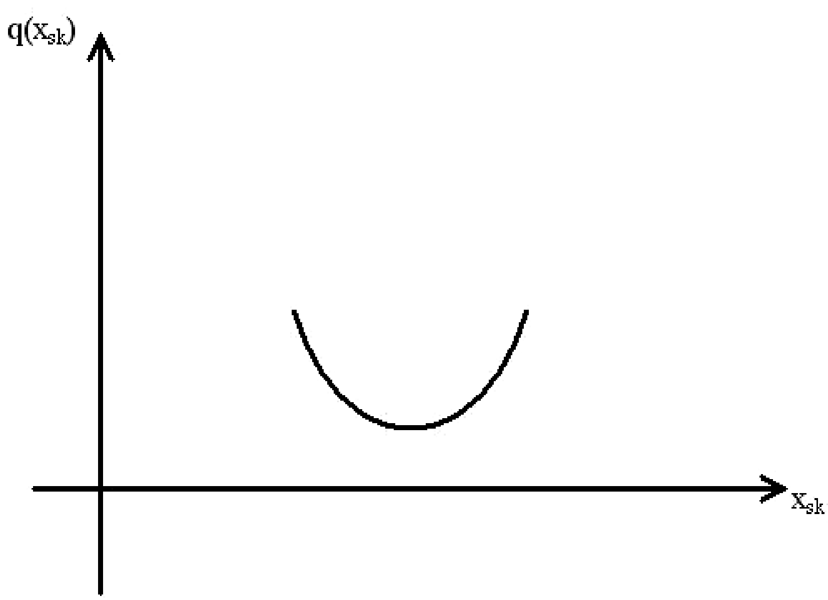
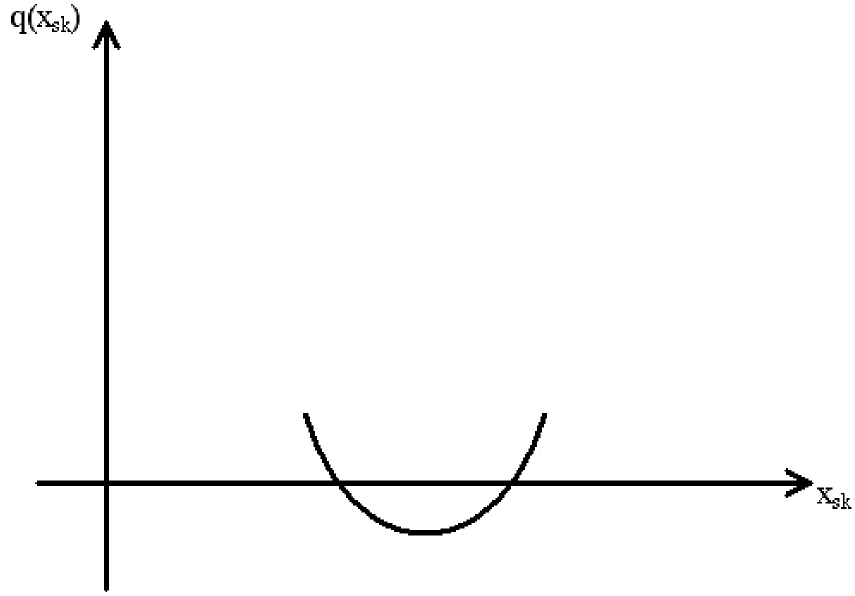


Figure 2.2.4: Case 4: a>0 and b²-ac>0



When there is no block effect, Graybill (1976) showed that Case 4 holds and a $1\text{-}\alpha$ confidence interval for x_{sk} exists if and only if $a > 0$, which is equivalent to rejecting the hypothesis H_o: $\beta_1 = 0$ in favor of H_a: $\beta_1 \neq 0$ with type I error rate α.. Graybill's algorithm, therefore, first carries out the test for zero slope and terminates without yielding a confidence interval for x_{sk} if H_o is not rejected. The situation is more complicated for the problem studied here where a block effect may exist, and '$a > 0$' does not guarantee that the discriminant, $b^2\text{-}ac$, is positive. Nonetheless, the calibration problem is only meaningful if $\beta_1 \neq 0$ and we also use a two stage procedure where we first test H_o: $\beta_1 = 0$ vs. H_a: $\beta_1 \neq 0$. We use the asymptotic normality of REML estimators and reject H_o with nominal type I error rate α if $\left|\hat{\beta}_1\right| / se\left|\hat{\beta}_1\right| \geq z_{\alpha/2}$. If H_o is rejected and b^2-$ac>0$, we construct an inverse prediction interval for x_{sk} with lower limit $\frac{-b-\sqrt{b^2-4ac}}{2a}$, and upper limit $\frac{-b+\sqrt{b^2-4ac}}{2a}$. Method 1 will use the 1-α quantile from an F distribution to evaluate a, b, and c. Method 2 will use the algorithm described below based on a bootstrap estimation of $\tilde{F}$ to evaluate a, b, and c.

2.3: Introduction to the Bootstrap

Using a bootstrap to approximate the distribution of a statistic is common practice. See for example Efron and Tibshirani (1998) where it is shown that an approximate confidence interval for a parameter can be obtained by using the percentiles of the bootstrap distribution of an appropriate pivotal. An assumption of the simple bootstrap is that the observations are independently and identically distributed. When this assumption holds, the bootstrap can be executed by sampling randomly with replacement from the observed data. Ideally, when bootstrapping the distribution of a pivotal, it is preferable to have a large number of bootstrap repetitions. Common practice is to carry out at least 1000 repetitions. Because of time limitations, in our study it was necessary to limit the number of bootstraps to 150.

2.4: Bootstrap Algorithm for Estimating $\tilde{F}_{0.95}$

Suppose we have observed or generated data **D** = $\{(X_{tj},Y_{tj}), t=1, \ldots, K; \; j=1, \ldots, n_t\}$ described in section 1.1 according to the algorithm given in section 3.3. And suppose we have rejected the hypothesis H_o: $\beta_1 = 0$ in favor of Ha: $\beta_1 \neq 0$ at nominal type I error rate.

Step 1*: Using a random number generator we simulated $\left\{\varepsilon^*_{tj}\right\} iid\ N(0,\hat{\sigma}^2_{\varepsilon})$ and independently $\left\{\eta^*_t\right\} iid\ N(0,\hat{\sigma}^2_{\eta})$.

Step 1a*: Independently, also generate $\hat{e}^*_{sk} \sim N(0,\hat{\sigma}^2_{\eta}+\hat{\sigma}^2_{\varepsilon})$ and store for step 3b*

Step 2*: To create the bootstrap data we will use the errors we simulated in step 1* and the REML estimators $\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}_{\varepsilon}, \hat{\sigma}_{\eta}$ to obtain data:

$$D^* = \{(x_{tj}, y^*_{tj} = \hat{\beta}_o + \hat{\beta}_1 x_{tj} + \varepsilon^*_{tj} + \eta^*_t),\ t = 1,2,\ldots,T,\ j = 1,2,\ldots,n_t\} \quad \textbf{(2.7)}$$

Step 3*: From data D^* using PROC MIXED of SAS we find estimators $\hat{\beta}_0^*, \hat{\beta}_1^*, \hat{\sigma}_\varepsilon^*, \hat{\sigma}_\eta^*$.

Step 3a*: We test H$_o$: $\hat{\beta}_1$=0. If H$_o$ is rejected, go to 3b*. Otherwise, no interval is obtained and return to bootstrap step 1*.

Step3b*: Using 1a*, estimate x_{sk} by

$$\hat{x}_{sk} = \frac{(y_{sk} - \hat{\beta}_0 - \hat{e}_{sk})}{\hat{\beta}_1} \tag{2.8}$$

Step 4*: Compute

$$Y_{sk}^* = \hat{\beta}_0^* + \hat{\beta}_1^* \hat{x}_{sk} + \varepsilon_{sk}^* + \eta_s^* \tag{2.9}$$

Step 5*: Compute scaled $\tilde{F}^*$ to simplify the notation, we will denote this as F* given by

$$\begin{aligned} F^* &= \frac{n}{n-2}\left[\frac{Y_{sk}^* - \hat{\beta}_0^* - \hat{\beta}_1^* \hat{x}_{sk}}{\sqrt{\hat{V}ar^*(Y_{sk}^* - \hat{\beta}_0^* - \hat{\beta}_1^* \hat{x}_{sk})}}\right]^2 \\ &= \frac{n}{n-2}\left[\frac{Y_{sk}^* - \hat{\beta}_0^* - \hat{\beta}_1^* \hat{x}_{sk}}{\sqrt{\hat{\sigma}_\varepsilon^* + \hat{\sigma}_\eta^* + \hat{V}ar(\hat{\beta}_0^*) + \hat{x}_{sk}^2 \hat{V}ar(\hat{\beta}_1^*) + 2\hat{x}_{sk}\hat{C}ov(\hat{\beta}_0^*, \hat{\beta}_1^*)}}\right]^2 \end{aligned} \tag{2.10}$$

Independently steps 1-5 are repeated 150 times, yielding $\left\{F_j^*\right\}_{j=1}^{150}$, which will then be arranged in increasing order: $F_{(1)}^* \le F_{(2)}^* \le ... \le F_{(150)}^*$. These order statistics can be used to approximate the 0.95 percentile of the distribution of $\tilde{F}$

An approximate 0.95 inverse prediction interval for X_{sk} is then given by

$$\frac{-b \pm \sqrt{4b^2 - 4ac}}{2a}$$

where

$$a = \hat{\beta}_1^{\,2} - \hat{V}ar(\hat{\beta}_1) f_{0.95}^*$$
$$b = \hat{\beta}_0 \hat{\beta}_1 - \hat{\beta}_1 Y_{sk} - 2\hat{C}ov(\hat{\beta}_0, \hat{\beta}_1) f_{0.95}^*$$
$$c = Y_{sk}^2 - 2\hat{\beta}_0 Y_{sk} - f_{0.95}^* (\hat{\sigma}_{\varepsilon} + \hat{\sigma}_{\eta} + \hat{V}ar(\hat{\beta}_1)$$

Chapter 3 – Simulation Study

3.1: Overview of Simulation Study

Our simulation investigates the performance of our two methods in constructing inverse confidence intervals. The investigation was carried out by simulating data that follow the model in equation (1.1) using a variety of settings. We then bootstrapped the distribution of $\tilde{F}$ using the algorithm described in section 3.4 and constructed approximate 95% inverse prediction intervals for $x_{sk.}$ using the 0.95 quantile of the bootstrapped distribution and the 0.95 quantile of an F distribution. The performance of both methods was examined by measures such as coverage rate and average length of the confidence intervals. Our simulation was run in the statistical software SAS 9.1. Summary tables were made with Excel and figures of the results were made with the statistical software R as well as SAS 9.1.

3.2: Fitting the Model

The model from which we generated our data from can be expressed in matrix notation as

$$\boldsymbol{Y} = X\boldsymbol{\beta} + \boldsymbol{e}, \qquad \textbf{(3.1)}$$

The way in which we generate the independent variables ensures that the design matrix X has full rank with probability 1 and **e**~N(0, V) where V=(e_{tj}), where e_{tj} is defined as in (1.2). If we define y_{sk} to be a new observation of the response Y, the covariance matrix K is given by

$$K = Cov(e_t, e_s) = \begin{Bmatrix} \sigma_\eta^2 J + \sigma_\varepsilon^2 I & for\ t = s \\ 0 & for\ t \neq s \end{Bmatrix} \qquad \textbf{(3.2)}$$

where J is a matrix of 1's and I is the identity matrix, both having n_t rows and n_t columns. Because we want to predict x_{sk} given $Y=y_{sk}$ it was necessary to generate an x_{sk} so that we would have a way to compare our methods based on how the prediction intervals captured x_{sk}. We chose to randomly generate x_{sk} from a Uniform (0, 1) distribution using the SAS RANUNI command.

3.3: Data Generation

Simulations were run in SAS using proc mixed for analysis and proc SQL for data manipulation. SQL statements were needed in order to manipulate data properly, and join data sets together so that computations could be handled easier. Seed generation for each simulation was done using a random uniform number on (0, 1) and multiplying that number by 1×10^8, and truncating the result.

Steps used to generate data **D**.

Step 1: Generate η_t from N(0, σ_η), and independently $\{\varepsilon_{tj}\}$ from N(0, σ_ε).

Step 2: Generate x_{tj} from a Uniform(0,1) distribution.

Step 3: Let

$$Y_{tj} = \beta_0 + \beta_1 x_{tj} + e_{tj},$$

where

$$e_{tj} = \eta_t + \varepsilon_{tj}.$$

Step 4: Generate x_{sk} from a Uniform (0,1) (store it for later).

Step 5: Independently we generate η_s from N(0, σ_η), and ε_{sk} from N(0, σ_ε).

Step 6: Compute

$$Y_{sk} = \beta_o + x_{sk}\beta_1 + e_{sk}$$

where,

$$e_{sk} = \eta_s + \varepsilon_{sk}$$

Step 7: Use PROC MIXED of SAS to estimate β_o ,β_1,σ_ε, *and* σ_η , store for later use.

Step 8: Test H_o: β_1=0, if Ho is rejected we will not form a confidence interval for that replication of the simulation. We carry out the bootstrap only on the cases where H_o: β_1 =0 is rejected.

3.4: Simulation Settings

Simulation settings where chosen to see how coverage rates varied over different parameter values. Three different settings for β_1 were chosen, these varied from low, medium, and high. The parameter β_0 was set to 0 for all simulations.

For the time variance, σ_η, and location variance, σ_ε, settings were chosen so that the ratio of these quantities varied from low, medium and high. In the low ratio setting the location variance has a higher setting then the time variance. In the medium ratio setting, the ratio of the variances is equal. In the high ratio setting the time variance has a higher setting then the location variance. These settings were chosen in relation to the analysis of a split-plot design, and how well these tests perform due to the ratio of whole plot to sub plot error.

Time settings and location settings were chosen in a similar manner having both high and low settings. Two other settings, the number of replications of a given simulation setting, and the number of bootstraps per unique simulation setting had to be set in conjunction with the time and location settings so that a specific simulation could be completed in a reasonable amount of time. Thus settings for the factors time and location were chosen with only two levels, low and high. This was necessary to ensure that simulations would be able to be completed on time. A total of 36 different simulations were run. Below is a table with the settings discussed above:

Table 3.4.1: Parameter Settings

	Low	**Medium**	**High**
β_o	0	0	0
β_1	2	8	20
$\sigma_\eta^2 / \sigma_\varepsilon^2$	0.05/5	5/5	5/0.05
Time=t	8	-	12
Location=j	6	-	20

Number of Bootstraps Replications: 150

Number of Replicated Simulations: 200

3.5: Simulation ID

In order to identify the different simulations a 5-digit character identifier was adopted for each simulation. The first digit represented the settings for the slope in our model: 1 = low (2), 2 = medium (8), and 3 = high (20). The second digit represented the settings for the ratio of the time variance to the location variance: 1=low (time .05/loc 5), 2=med (time 5/loc 5), and 3= high (time 5/loc .05). The third digit represented the amount of times in our model: 1=low (8), and 2=high (12). The fourth digit represented the amount of locations in our model: 1=low (6), and 2=high (20). The fifth digit was the value of the intercept and for this simulation study always had a default of zero.

An example of a simulation id would be: 12120. This id can be interpreted as having the following configuration: The first digit represents the slope and it has a value of 1 so the slope of our simulated model has been given the low setting (2). The second digit represents the ratio of time variance and location variance. The second digit has a value of 2 so the ratio of time variance over location variance has been given the medium setting (5/5). The third digits represents the number of times, it has a value of 1, which tells us that time has been given the low setting (8). The fourth digit represents the number of locations; and has a value of 2 so location has been given the high setting (20). The fifth digit represents the intercept, and for the purposes of this report will always be given the value of 0.

Chapter 4 - Results

4.1: Introduction to Results Chapter

We conducted a simulation study to compare the performance of the two methods described in section 1.1. These methods were used to obtain inverse prediction intervals for x_{sk} in the mixed model setting given in equation (1.1). Evaluative measures such as coverage rate, and average length and standard deviation of the interval lengths were used to compare the two methods. Additionally, McNemar's test was carried out using PROC FREQ in SAS 9.1 to determine if the methods were performing the same for each distinct simulation setting. Tables of the simulation results used to create figures in sections 4.2, and 4.3 are in appendix B, and additional figures related to results described in section 4.2 are in appendix C. Each 2x2 table created using PROC FREQ and McNemar's test can be found in appendix D.

Keep in mind that "Method 1" refers to the method that uses $f_{0.95}$, the 95th quantile of an F distribution, to form approximate 95% inverse prediction sets for x_{sk}, "Method 2" refers to the method that uses $f^*_{0.95}$ from the bootstrapped distribution of $\tilde{F}$ to form approximate 95% inverse prediction sets for x_{sk}. The sections that follow will make use of the unique simulation identifier denoted 'simulation ID' that was defined in section 3.5.

4.2: Average Interval Width and Standard deviation

Using the table of data in appendix B, whisker plots were made for each unique simulation and grouped by β_1 settings (low=2, medium=8, high=20) and $\sigma_\eta / \sigma_\varepsilon$ settings (low=0.05/5, medium=5/5, high=5/0.05). These figures are placed in Appendix D. The dot represents the mean length and whiskers extend to 1.96 times the standard error of the interval length, so that what are represented are 95% confidence intervals for the mean interval length.

The lengths for both methods are very large for the first twelve cases relative to the target values x_{sk}, which lies in the unit interval. Those 12 cases coincide with the 12 simulations where β_1 was set to low. Using the SAS procedure, PROC UNIVARIATE, a two-sided sign test was performed to compare mean interval length between method 1 and method 2 across the 36 separate simulation settings. The test yielded a p-value of 0.003 indicating that the mean lengths of the two methods are statistically significantly different.

To further study interval length, Figures 4.2.1 – 4.2.4 present box plots of length plotted against an effect size type parameter Δ defined by

$$\Delta = \beta / \sqrt{\sigma_\eta^2 + \sigma_\varepsilon^2} \tag{4.1}$$

For small Δ, observing 'Y' conveys little information about the corresponding 'X' and the test for zero slope is expected to have low power when $\Delta > 0$, unless sample size is large. The non-decreasing values of Δ in our design are given in the next to last column of table 4.2.1. All labels with same first letter, A-F, have the same Δ value. As expected, ignoring sample size, interval lengths for both methods decrease with increasing Δ. The intervals are very wide for the first twelve cases, where Δ is smallest. Plots for both methods of mean length in figure 4.2.5 and median 'mean length' vs. Δ in figure 4.2.6 convey the same information. As in previous graphs where both methods were plotted red circles represent results based on method 1 and black circles represent results based on method 2. Note that in figure 4.2.5 and 4.2.6 it appears that mean length for method 2 tends to be higher than for method 1 for small values of Δ.

Table 4.2.1: Simulation ID with corresponding Δ

Sim Id	β_1	$\sigma_\eta^2 / \sigma_\varepsilon^2$	$\beta_1 / \sqrt{\sigma_\eta^2 + \sigma_\varepsilon^2} = \Delta$	Label given in figures
12110	2.00	med	0.632456	A12110
12120	2.00	med	0.632456	A12120
12210	2.00	med	0.632456	A12210
12220	2.00	med	0.632456	A12220
11110	2.00	low	0.889988	B11110
11120	2.00	low	0.889988	B11120
11210	2.00	low	0.889988	B11210
11220	2.00	low	0.889988	B11220
13110	2.00	high	0.889988	B13110
13120	2.00	high	0.889988	B13120
13210	2.00	high	0.889988	B13210
13220	2.00	high	0.889988	B13220
22110	8.00	med	2.529822	C22110
22120	8.00	med	2.529822	C22120
22210	8.00	med	2.529822	C22210
22220	8.00	med	2.529822	C22220
21110	8.00	low	3.559953	D21110
21120	8.00	low	3.559953	D21120
21210	8.00	low	3.559953	D21210
21220	8.00	low	3.559953	D21220
23110	8.00	high	3.559953	D23110
23120	8.00	high	3.559953	D23120
23210	8.00	high	3.559953	D23210
23220	8.00	high	3.559953	D23220
32110	20.00	med	6.324555	E32110
32120	20.00	med	6.324555	E32120
32210	20.00	med	6.324555	E32210
32220	20.00	med	6.324555	E32220
31110	20.00	low	8.899883	F31110
31120	20.00	low	8.899883	F31120
31210	20.00	low	8.899883	F31210
31220	20.00	low	8.899883	F31220
33110	20.00	high	8.899883	F33110
33120	20.00	high	8.899883	F33120
33210	20.00	high	8.899883	F33210
33220	20.00	high	8.899883	F33220

Figure 4.2.1: Box plots of method 1 interval lengths vs. Δ (A-B)

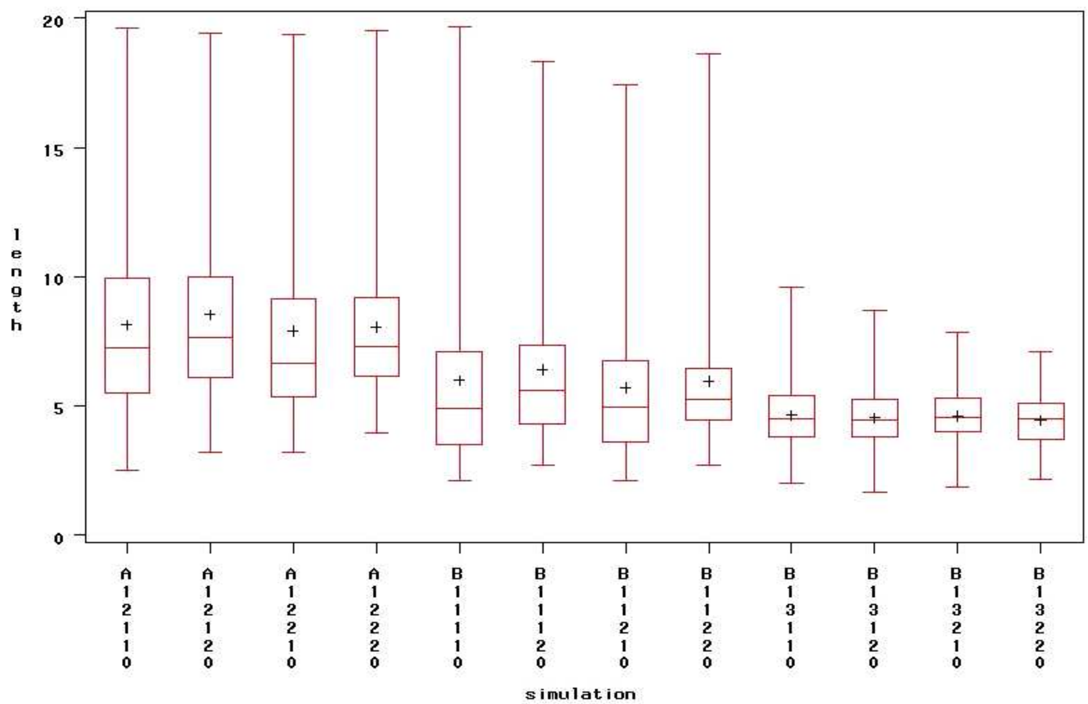


Figure 4.2.2: Box plots of method 2 interval lengths vs. Δ (A-B)

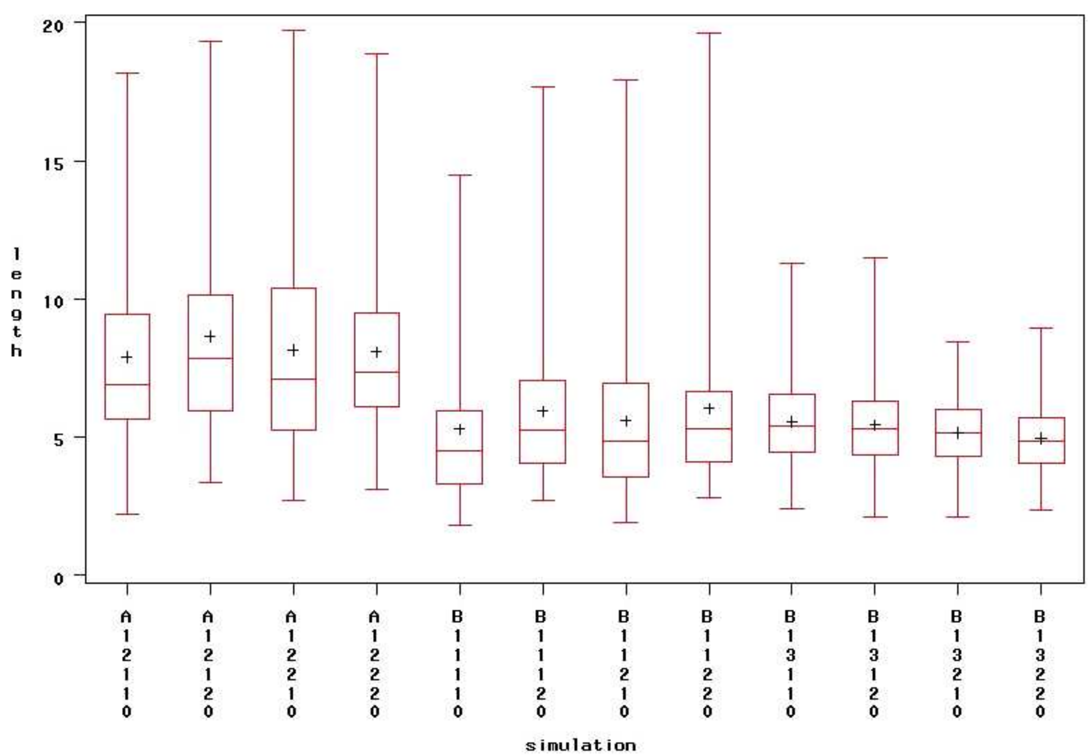


Figure 4.2.3: Box plots of method 1 interval lengths vs. Δ (C-F)

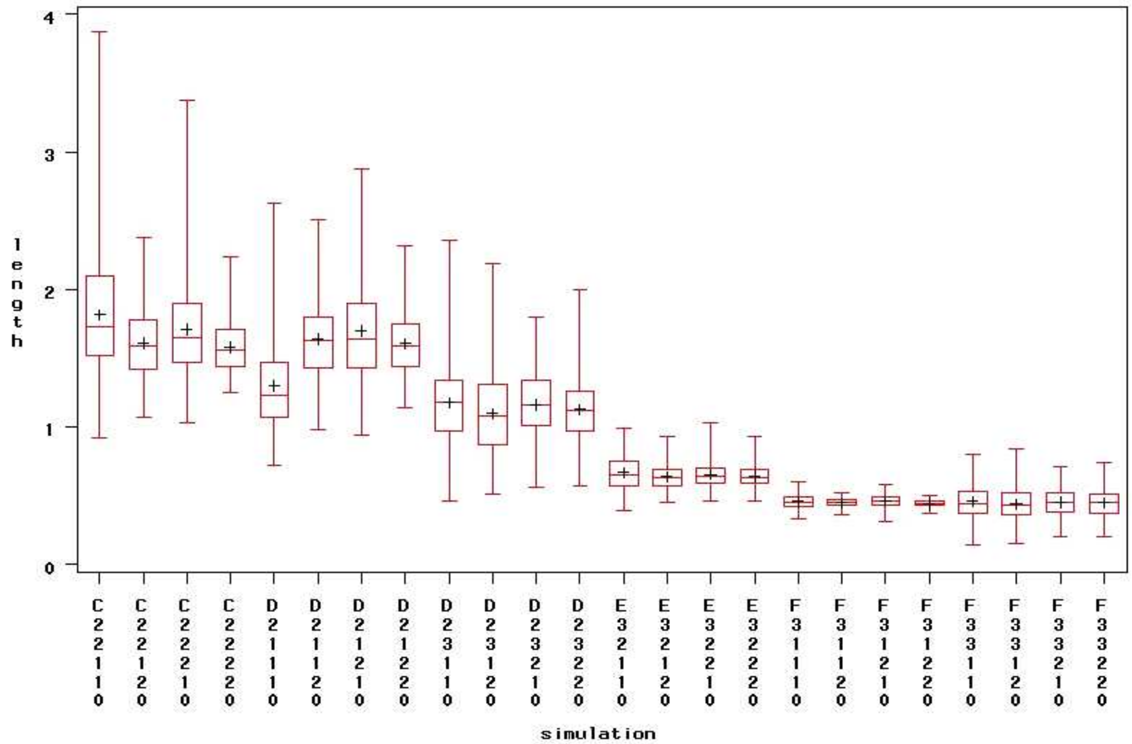


Figure 4.2.4: Box plots of method 2 interval lengths vs. Δ (C-F)

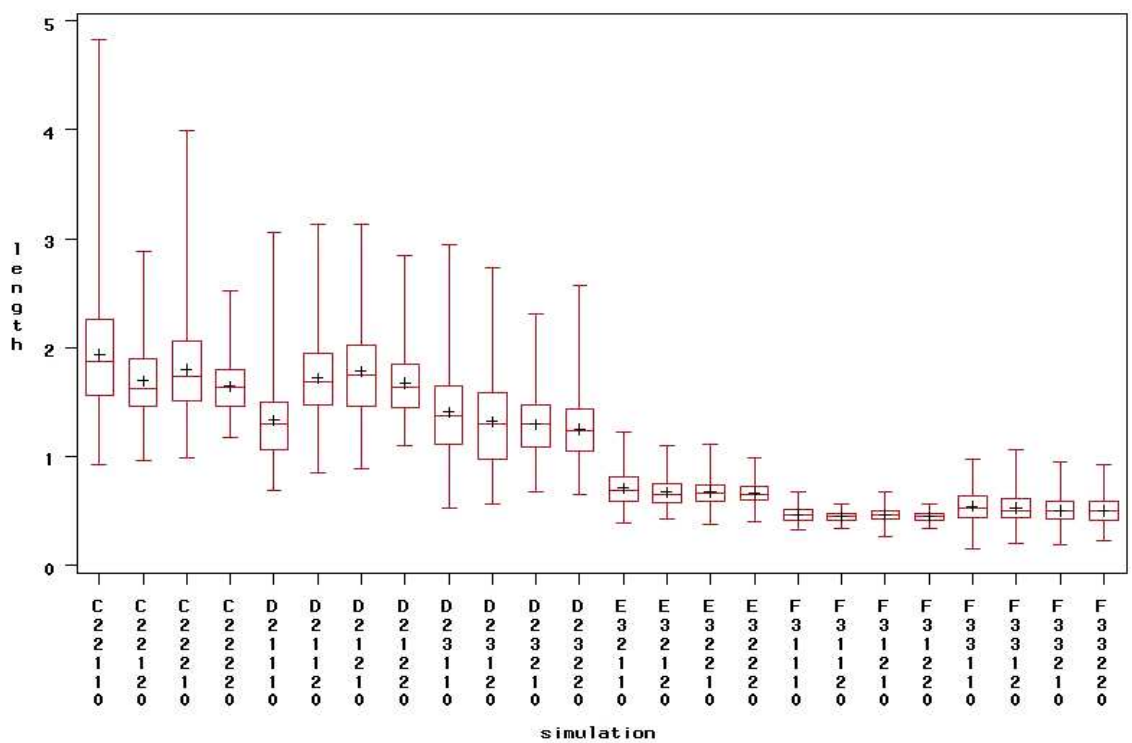


Figure 4.2.5: Plot of mean length vs. Δ

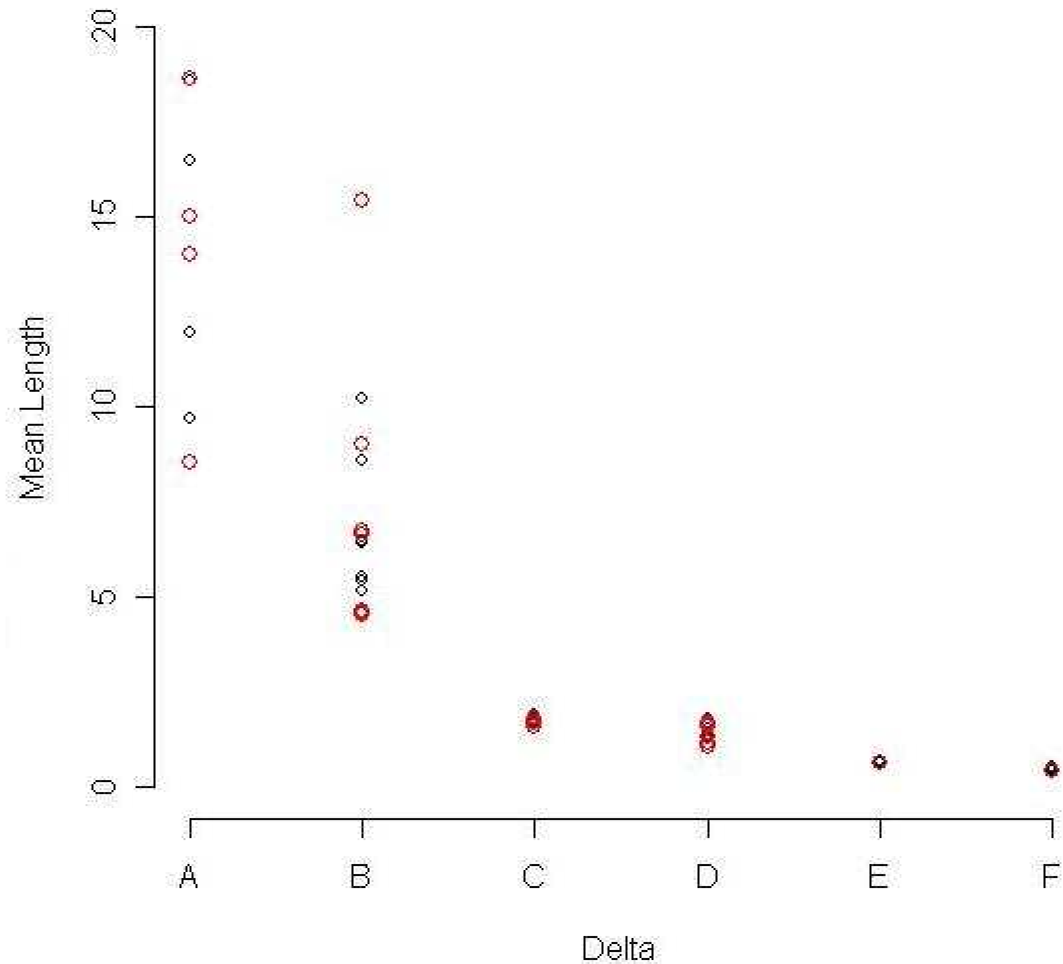
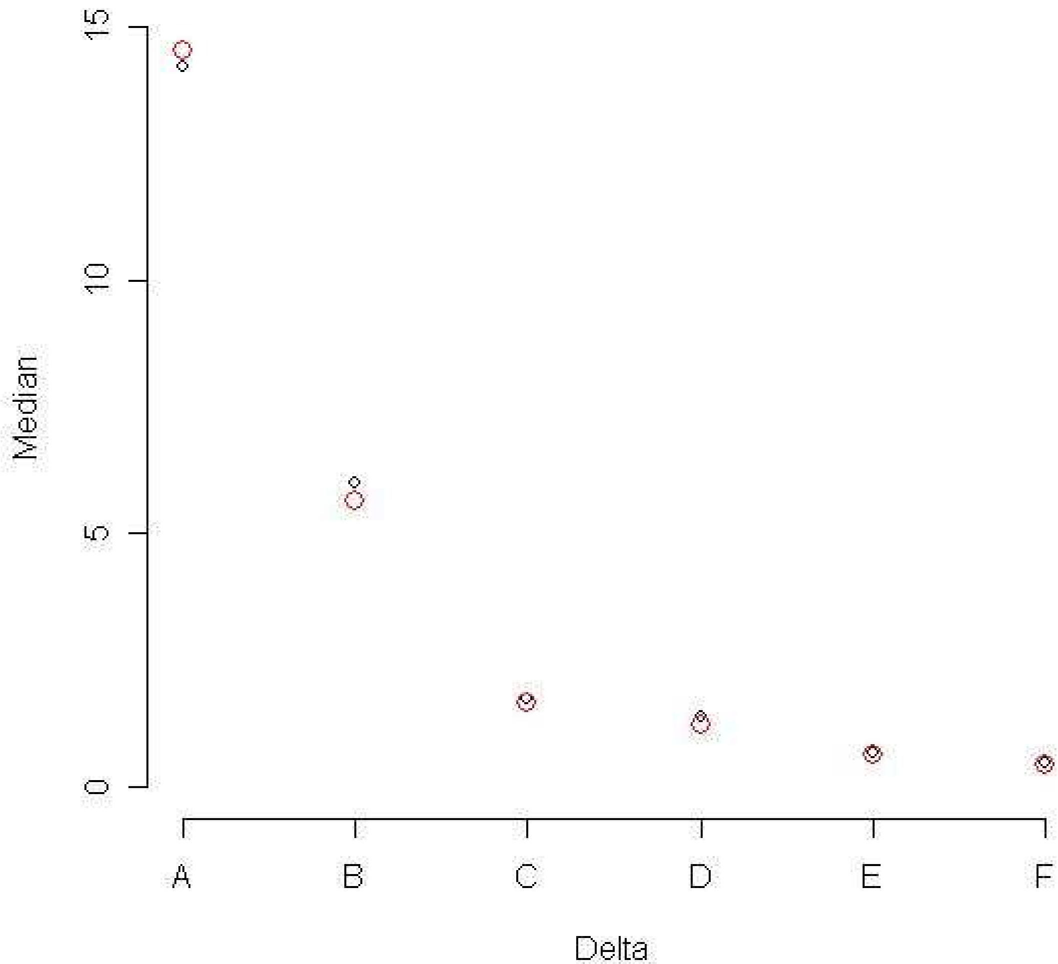


Figure 4.2.6: Plot of median mean length vs. Δ



4.3: Coverage Rate

Table (4.3.1) presents estimated coverage rates of nominal 0.95 confidence intervals for x_{sk} based on those data sets for which intervals could be constructed. The standard errors of these estimates vary among the Simulation ID's since the number of sets where S is an interval varies among the parameter settings (see section 4.4). As a rough guide, $\sqrt{(.89)(.11)/100} = 0.031$ provides an approximate upper bound on these standard errors. Cases where a 95% confidence interval for coverage rate contains the target rate of '0.95' are indicated in bold. Method 2 appears to have more simulations where the coverage rate is captured by the 0.95 confidence interval then does method 1. However, overall both methods appear to have coverage rates below nominal for about 1/3 of the simulation settings.

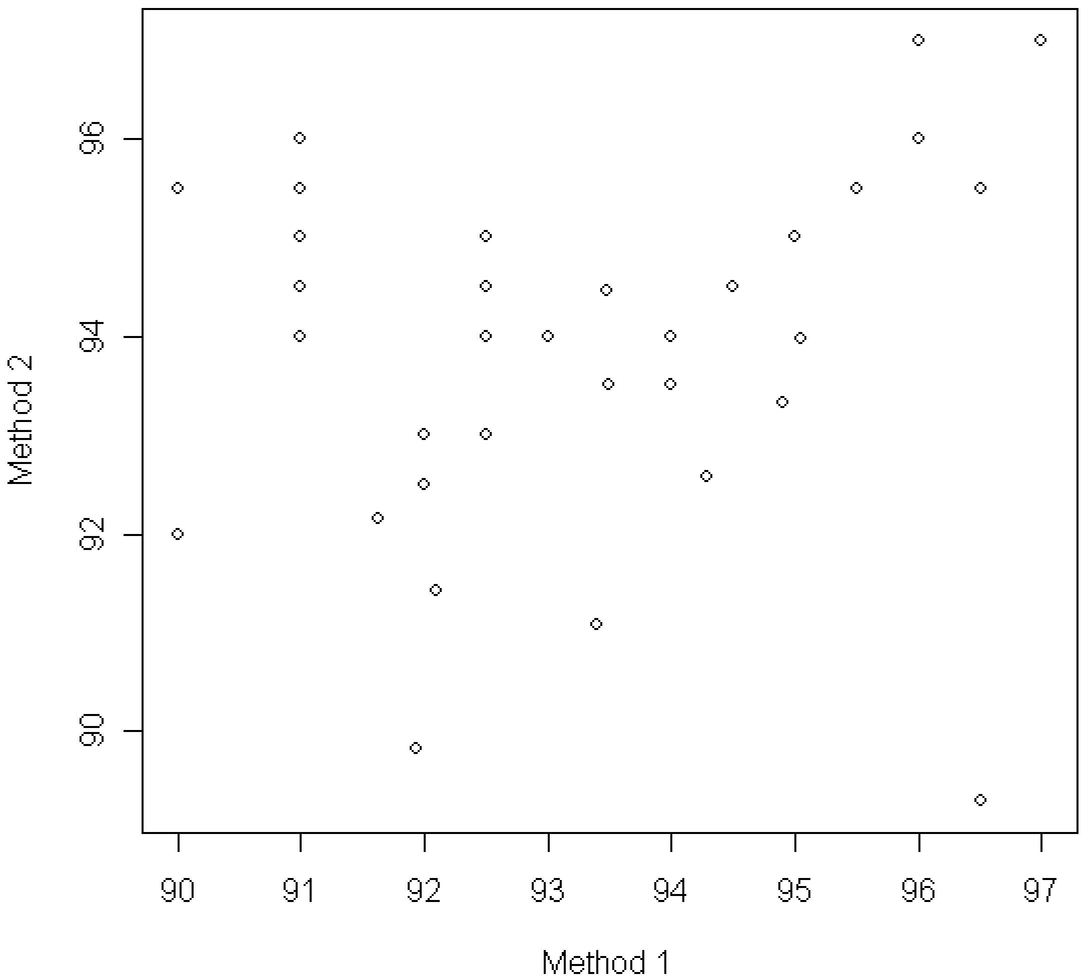
McNemar's test for the equality of two correlated proportions was used to test for a difference between the coverage rates across the cases that were investigated. The estimated rates are correlated since both methods were used on the same data set generated under each setting. P-values from McNemar's test are given in Table (4.4.1). From this table we see that with the exception of a few simulations, the simulations where McNemar's test found a significant difference between the two methods are those where the variance ratio is set to high. This signifies that the performance of the two methods is significantly different in terms of coverage rate as long as the time variance component is large relative to the location variance component.

Table 4.4.1: Simulation coverage rates for method 1 and method 2

Simulation ID	Method 1	Method 2	P-value
11110	**96.51**	89.29	0.014
11120	**94.29**	92.57	**0.180**
11210	**95.04**	**93.97**	**0.564**
11220	**94.90**	93.33	**0.180**
12110	91.94	89.83	**0.317**
12120	92.09	91.43	**0.655**
12210	**93.40**	91.09	**0.317**
12220	91.62	92.15	**0.655**
13110	91.00	**94.00**	0.014
13120	90.00	92.00	0.046
13210	90.00	92.00	0.046
13220	92.50	**94.00**	0.083
21110	**96.50**	**95.50**	**0.157**
21120	**96.00**	**96.00**	**1.000**
21210	**95.50**	**95.50**	**1.000**
21220	92.00	93.00	**0.317**
22110	92.50	**94.00**	**0.083**
22120	92.50	**94.50**	**0.103**
22210	**97.00**	**97.00**	**1.000**
22220	**95.00**	**95.00**	**1.000**
23110	91.00	**96.00**	0.002
23120	91.00	**94.50**	0.008
23210	**93.47**	**94.47**	**0.157**
23220	92.00	92.50	**0.317**
31110	**94.00**	**93.50**	**0.564**
31120	**94.00**	**94.00**	N/A
31210	**94.50**	**94.50**	**1.000**
31220	**96.00**	**97.00**	**0.157**
32110	**93.50**	**93.50**	**1.000**
32120	**96.00**	**97.00**	**0.157**
32210	93.00	**94.00**	**0.317**
32220	92.50	93.00	**0.564**
33110	91.00	**95.00**	<.00001
33120	90.00	**95.50**	0.001
33210	92.50	**95.00**	0.025
33220	91.00	**95.50**	0.003

Figure 4.4.1 plots the coverage rates for the two methods given in Table 4.4.1 against one another. Here, we see little relation between the rates for the two methods.

Figure 4.4.1: Plot of coverage rate for method 1 vs. coverage rate for method 2



Coverage rates for both methods are plotted against the slope β_1 in Figure 4.4.2 against variance ratio in Figure 4.4.3, against time in Figure 4.4.4 and against location in Figure 4.4.5. Black Circles Represent results based on method 2 and red circles represent results based on method 1. From these plots we see that, overall, method 1 appears to have a higher coverage rate then method 2 when the slope (β_1) is set to low (2), and method 2 appears to have a higher coverage rate then does method 1 when the variance ratio ($\sigma_\eta^2 / \sigma_\varepsilon^2$) is high (5/0.05).

Figure 4.4.2: Plot of coverage vs β_1 for method 1 and method 2

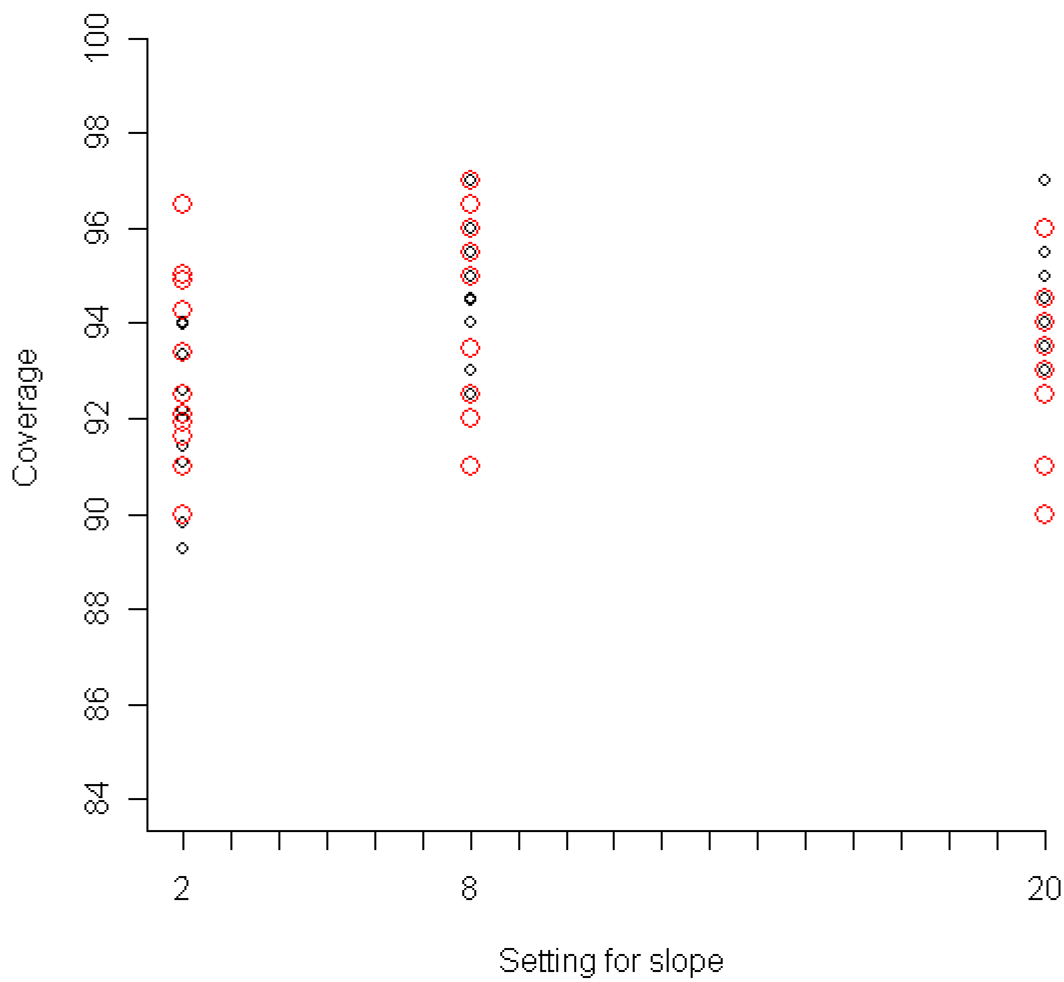


Figure 4.4.3: Plot of coverage vs. $\sigma_\eta / \sigma_\varepsilon$ for method 1 and method 2

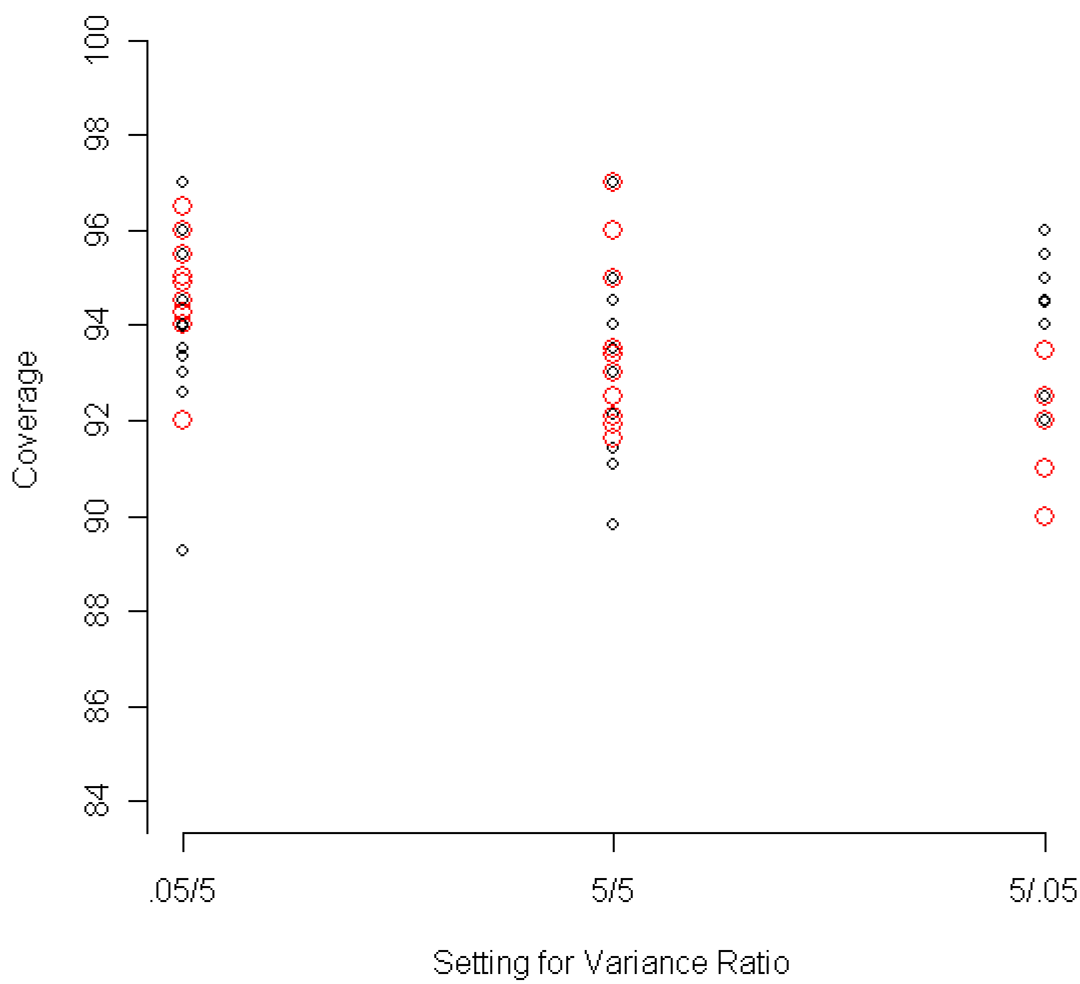


Figure 4.4.4: Plot of coverage vs. time for method 1 and method 2

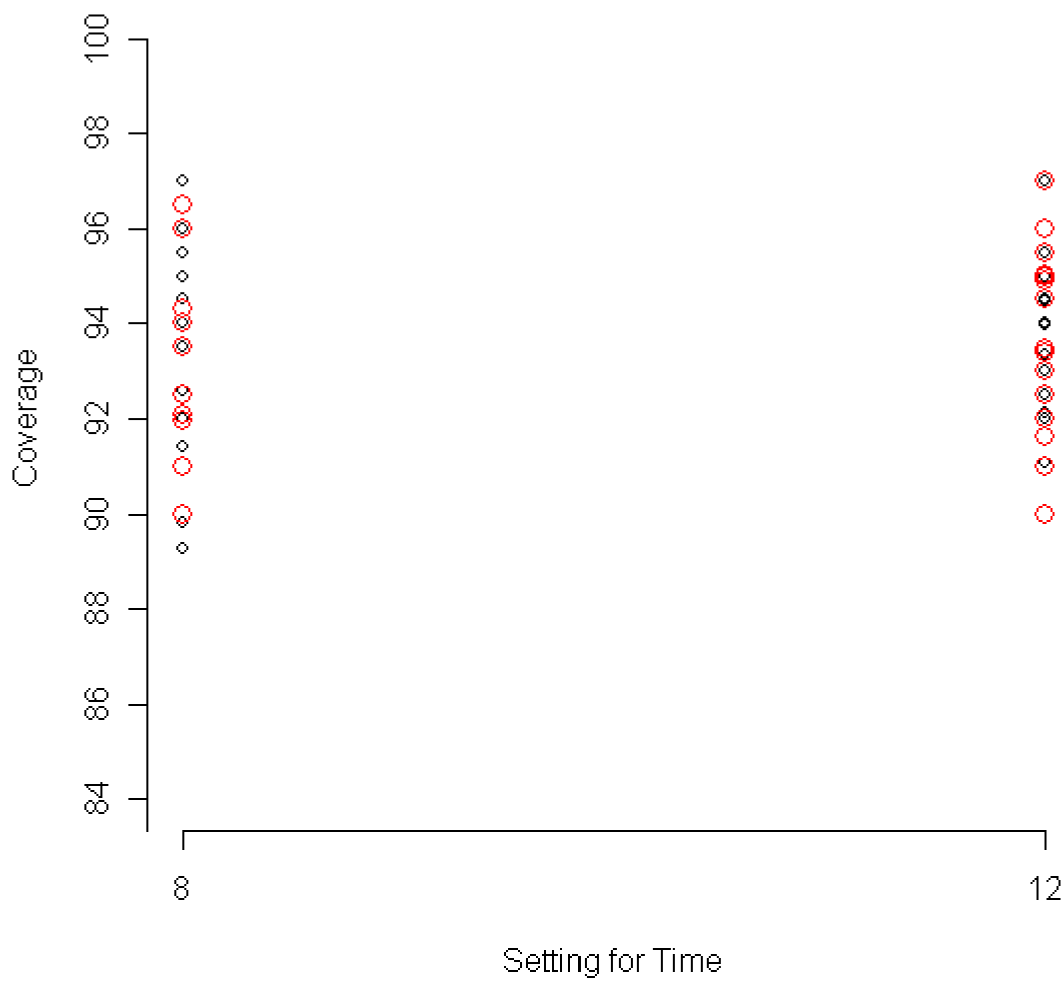


Figure 4.4.5: Plot of coverage vs. location for method 1 and method 2

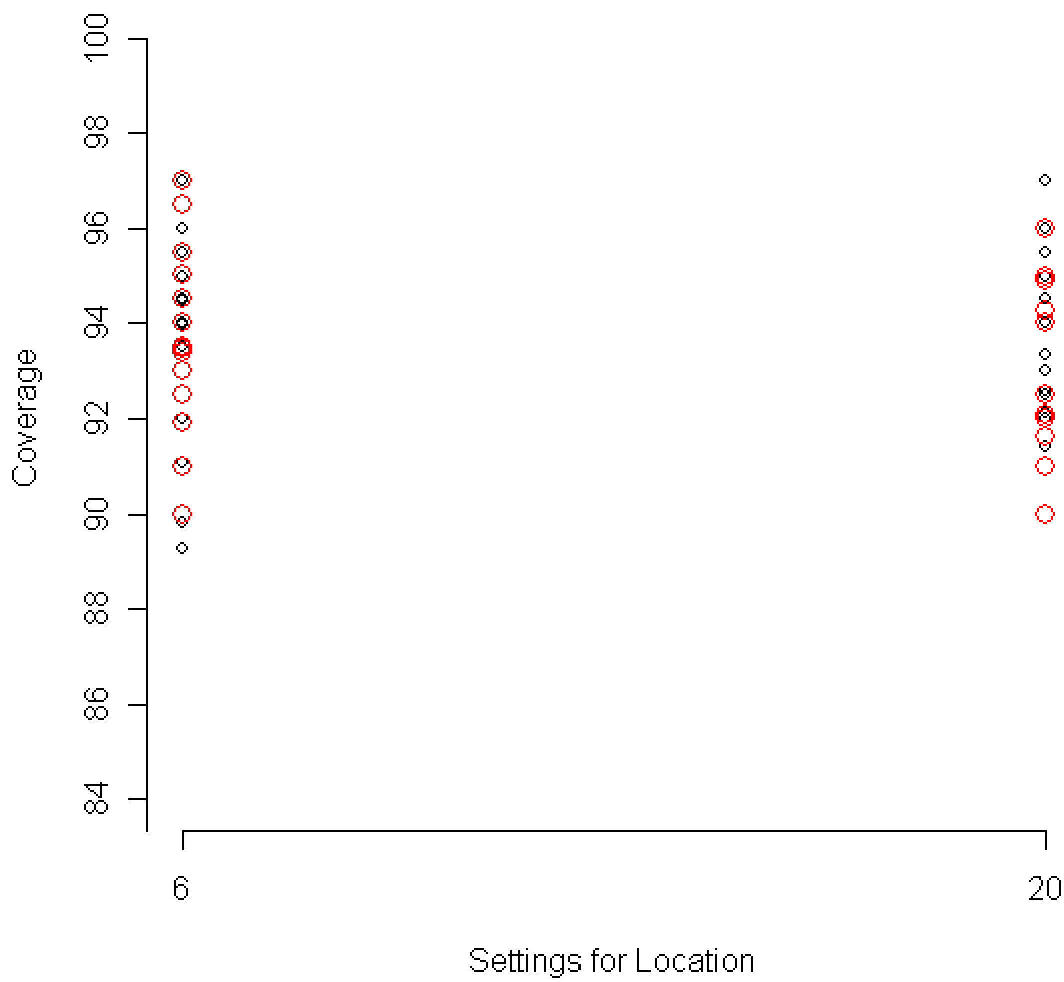
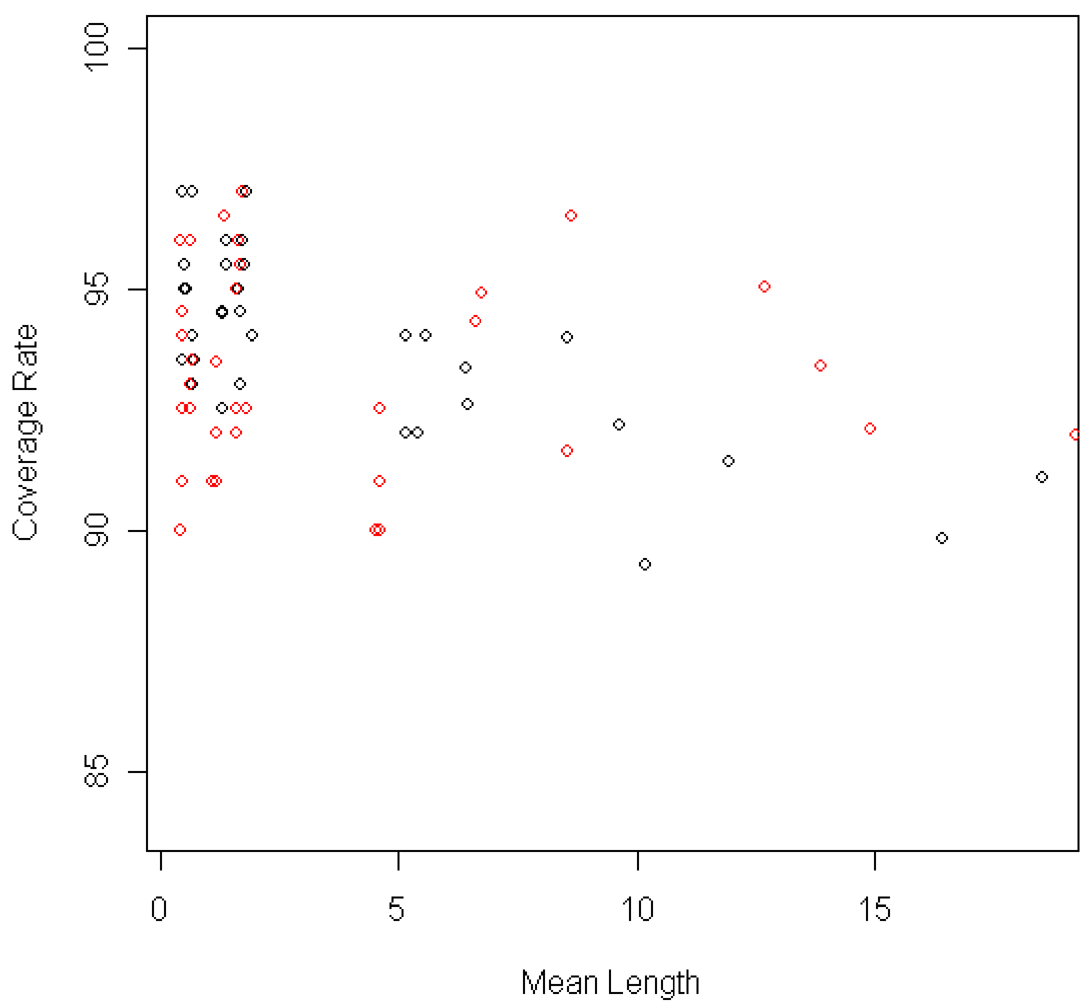


Figure 4.4.6 plots coverage rate vs. mean length. There appears to be a slight downward trend in coverage rate as mean length increases, but this could be due to the low slope setting for those simulations and the resulting smaller number of confidence intervals that could be constructed due to the slope being zero. Other then this there appears to be little relation between these variables for both methods.

Figure 4.4.6: Plot of coverage rate vs. mean length for method 1 and method 2



4.4: Bootstrap Diagnostics

Although the number of bootstraps used to approximate the distribution of $\tilde{F}$ was fixed to be 150, not every bootstrap was able to be carried to the final step. If the condition $\hat{\beta}_1 \neq 0$ from Step 3a* in Section 2.4 was not met, f* could not be computed. To determine, on average, how many f*'s were actually involved in approximating the distribution of $\tilde{F}$ for each unique simulation setting, the following measure was adopted.

Consider a given simulation setting, and a single inverse prediction interval i, i=1,…,200, within that setting. For that given simulation, and inverse prediction interval i, denote the number of bootstraps where $\hat{\beta}_1 \neq 0$ by M_i. The table below summarizes the average number of M_i's, denoted $\bar{M}$, and standard deviation of the M_i's, denoted S_M, for each unique simulation id.

Table 4.3.1: $\bar{M}, S_M$

Simulation ID	$\bar{M}$	s_M
11110	69.59	44.9697
11120	121.995	32.9077
11210	87.225	46.1277
11220	138.52	19.9136
12110	91.85	43.3037
12120	118.66	33.8763
12210	79.07	44.4412
12220	137.18	23.0223
13110	149.955	0.2307
13120	150	0
13210	149.935	0.2667
13220	150	0
21110	149.535	5.6059
21120	150	0
21210	150	0
21220	150	0
22110	149.77	0.8723
22120	150	0
22210	149.99	0.0997
22220	150	0
23110	149.935	0.2471
23120	150	0
23210	149.905	0.3113
23220	150	0
31110	150	0
31120	150	0
31210	150	0
31220	150	0
32110	150	0
32120	150	0
32210	150	0
32220	150	0
33110	149.975	0.1565
33120	150	0
33210	149.965	0.1842
33220	150	0

Note that most of the bootstraps where $H_o : \beta_1 = 0$ was not rejected were those where the slope, β_1 was set to low and the variance ratio was at the low and medium settings.

4.5: Simulation Diagnostics

Recall that both methods first test $H_0 : \beta_1 = 0$ and check to see if the discriminate is positive before attempting to construct a confidence interval for x_{sk}. Table (5.3.1) indicates that both methods fail at high rates to yield intervals when β_1 is at its low setting. This agrees with what we found in Table (4.3.1) when we summarized for which simulations $\hat{\beta}_1 \neq 0$ Additionally Table (4.4.1) shows that Method 1 did not have any intervals where the discriminant was negative, but Method 2 did give some confidence intervals where the discriminant was negative in spite of the fact that the slope was found to be nonzero for those intervals.

Table 4.4.1: Settings for which β_1=0 or b^2-ac<0

Sim Id	Total number of intervals that could not be formed Method 1	Total number of intervals that could not be formed Method 2	Intervals where slope non-zero, but b^2-ac<0 Method 1	Intervals where slope non-zero, but b^2-ac<0 Method 2	Intervals where Slope is 0
11110	114	116	0	2	114
11120	25	25	0	0	25
11210	79	84	0	5	79
11220	4	5	0	1	4
12110	76	82	0	6	76
12120	23	25	0	2	23
12210	94	99	0	5	94
12220	9	9	0	0	9
23210	1	1	0	0	1

Chapter 5 - Summary and Conclusion

This report proposed and studied a solution to the calibration or inverse prediction problem in a mixed model setting where experimental units were selected from blocks that are treated as random effects. Two methods for producing inverse prediction intervals for x_{sk} were compared. Method 1 made inverse prediction intervals in the same way as Graybill proposed for the simple linear model by using quantiles from an F distribution. Method 2 accounts for the block effect by using a bootstrap to approximate the distribution of $\tilde{F}$ and forms inverse prediction intervals for x_{sk} with quantiles from the bootstrapped distribution of $\tilde{F}$. While results from each method indicate that both methods maintain coverage rates below 0.95 for approximately 1/3 of the chosen simulation settings. method 2 appears to have a slightly better coverage rate then does method 1.

Overall, method 2 produced coverage rates for prediction intervals near ninety-five percent. Thus within the space of our parameter settings, one might prefer method 2 over method 1. However when the slope setting is set to low, we notice some problems with method 2's approach. Specifically, some prediction intervals fail to form due to the discriminate being negative. In this case method 1 is a better choice since this procedure never failed to yield an interval where the discriminate was negative. Other problems such as computer resources may limit the use of method 2, since the bootstrap algorithm used to produce the intervals must be performed on a machine with good resources, namely a fast processor. For a researcher with limited resources and time, method 1 might be the best choice; especially since the coverage rate of intervals constructed using method 1 is comparable to the coverage rate of intervals constructed using method 2. One must also take into consideration that method 1 makes a strong assumption by using a statistic from an F distribution, thus one can see the utility of method 2. While the author of this paper would recommend method 2 with some restrictions as stated, further studies should be carried out before one can say one method is 'better' than the other.

Additionally, investigation should be carried out into why, for certain settings, Method 2 produced intervals that could not be formed due to the discriminant being negative. Also, other simulations should be carried out with added settings for all the parameters, in particular the settings for number of locations and times, and the variance ratio. Limited resources were available for running our simulations, and thus settings had to be chosen accordingly. However, in a high performance computing environment (HPC) one could take advantage of clusters and run very fast simulations with additional settings for all parameters as well as higher location and time settings. The availability of HPC would give us a better understanding of the behavior of both methods, and might lead to a better theoretical understanding behind the performance the two methods.

References

Casella, G. and Berger, R. L. (1990). Statistical Inference. Wadsworth/Brooks Cole, Pacific Grove, CA.

Brown, G.H. (1979). An optimisation criterion for linear inverse estimation. ***Technometrics*** **21,** 575-579.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. ***Ann. Statist.*** **7**, 1-26.

Efron, B and Tibshirani, R. (1998). An Introduction to the Bootstrap (Monographs on Statistics and Applied Probability). CRC Press LLC, Boca Raton, FL.

Jiang, J. (1996). REML estimation: Asymptotic behavior and related topics. ***Ann. Statist.*** **24**, 255-286.

Graybill, F.A. (1976). Theory and Application of the Linear Model. Duxbury, United States.

Shukla. G.K. (1972). On the problem of calibration. ***Technometrics*****,** **14**:547-553.

Williams, E.J. (1969a). Regression methods in calibration problems. ***BUN. Int. Statbt. Inst.*** **43,** 17-28.

Williams, E.J. (1969b). A note on regression methods in calibration. ***Technometrics*** **11,** 189-192.

Xiao, Bohong (2000). The linear calibration problem.

APPENDIX A – SIMULATION CODE

The following is the simulation code I ran for my report. Following this is additional code I used to merge information from all simulations together, and perform additional analyses that were required.

```
/**********************************************************************

      Version history Documentation:

      04-01-07 Beginning step 1 and 2 have been verified.
            Verification:
                  Steps 3 and 4, must be properly verified by hand.
            Future/Current issues:
                  *Data clean up, clean up data not needed/or used
during process.
                  *variance formula needs to be computed from proposal.
                  *need to asses how SAS determines arbitrary
percentiles from a data sample.
                  *Denominator 0 issue will need to be adressed.
                  *Optimize sql statements IF needed.

      06-04-07s
            Verification:
                  Code had been verified.
            Future/Current issues:
                  Generating multiple real experiments is taken care
of.
                  See above.

    06-13-07
            Future issues:
                  Made good progress but now optimization is an issue
SQL,

      12-26-07
            Verification:
                  Have checked program for errors.All the data
            sets are producing what is desired.

***********************************************************************
********************/

/*time v and loc v represent the variances for time and location in the
"real" experiment*/


%macro do_simulation(
seed,
alpha,
num_t,
```

```
num_l,
intercept,
slope,
var_terror,
var_lerror,
num_stat,
num_reps,
ID);

libname storage 'C:\test_case';
run;

libname output 'C:\test_case';
run;


PROC PRINTTO LOG="c:\logfile.log";
run;

proc printto print="c:\output.out";
run;


%let seed_increment=1;
run;

/****************************************************/
/* This is the real life data set that we are using */
/* This will contain the estimated b_0 and b_1      */
data real_Dset;
good_b1hat=0;
do rep=1 to &num_reps;
      x_sk=ranuni(&seed);
      time_sk=sqrt(&var_terror)*rannor(&seed);
      loc_sk=sqrt(&var_lerror)*rannor(&seed);

      do t=1 to &num_t;
            time_error = 0+sqrt(&var_terror)*rannor(&seed);

            do j=1 to &num_l;
                  location_error = 0+sqrt(&var_lerror)*rannor(&seed);
                  x=ranuni(&seed);
                  bo=&intercept;
                  b1=&slope;
                  y=bo+b1*x+time_error+location_error;
                  y_sk=bo+b1*x_sk+time_sk+loc_sk;
            output;
            end;
      end;
end;

DROP time_sk loc_sk;
run;
```

```
/* Analyze the data set real_dset marking those ones with good=1 and
good=0 */

DM 'Log; Clear;output;clear;';
run;

ods exclude all;

proc mixed data=real_dset CL method=REML;
by rep;
class t j;
model y=x / cl ddfm=kr solution CovB;
random t;
ods output covparms=var_params CovB=Cov_fix solutionf=solution_fixed
tests3=type3test;
run;

ods exclude none;


/* Note that this step is redundant but it's an extra check.
   One should not see a good value of 0 anymore! Examine type3test
below if worried about it.
 */
data type3test;
set type3test;
good_b1hat=0;
if probf<=.05 then good_b1hat=1;
run;


data solution_fixed;
set solution_fixed;
if effect="Intercept" then effect="B_0hat";
if effect="x" then effect="B_1hat";
run;

data var_params;
set var_params;
if covparm="t" then covparm="time_v";
if covparm="Residual" then covparm="loc_v";
run;




/*********************************************************************
**********************/
/* Finding the covariance between B_0hat and B_1hat,
   finding the variance of B_1hat
   finding the variance of B_0hat*/

proc sql;
create table data_prep8a as
select
      col1 as var_b0hat,
      col2 as cov_b1b0hat,
```

```
      rep
from cov_fix
where row=1;
quit;

proc sql;
create table data_prep8b as
select
      col2 as var_b1hat,
      rep
from cov_fix
where row=2;
quit;

proc sql;
create table data_prep8c as
select a.*,
         b.*
from data_prep8a as a, data_prep8b as b
where a.rep=b.rep;




proc transpose data=solution_fixed out=solution_fixed2;
by rep;
ID Effect;
run;

proc transpose data=var_params out=var_parms2;
by rep;
ID covparm;
run;

/******* End real data set  *****/


/******** Begin step 1 ***********/


data step1;
do rep=1 to &num_reps;
do exp=1 to &num_stat;
      do t=1 to &num_t;
            time_error_star = 0+rannor(&seed);

            do j=1 to &num_l;
                  location_error_star =0+rannor(&seed);
            output;
            end;
      end;
end;
end;
run;


proc sql;
```

```
create table temp_step as
select distinct
      a.*,
      b.time_v,
      b.loc_v
from step1 as a, var_parms2 as b
where
a.rep=b.rep AND
b._name_="Estimate";
quit;


data step1;
set temp_step;
time_error_star=time_error_star*sqrt(time_v);
location_error_star=location_error_star*sqrt(loc_v);
DROP time_v loc_v;
run;





proc sql;
create table step1a as
select a.*,
         b.x,
         b.x_sk,
         b.y_sk,
         c.B_0hat,
         c.B_1hat,
         d.time_v,
         d.loc_v,
         e.good_B1hat
from step1 as a, real_dset as b, solution_fixed2 as c,var_parms2 as d,
type3test as e
where a.t=b.t AND a.j=b.j
AND c._name_="Estimate"
AND d._name_="Estimate"
AND a.rep=b.rep
AND a.rep=c.rep
and a.rep=d.rep
and a.rep=e.rep;
quit;

proc sort data=step1a;
by rep exp;
run;

proc datasets library=work;
delete solution_fixed2 var_parms2 step1 temp_step;
run;
quit;



/***** begin step 2 ************************************/
```

```
/* Step 2 needed no modification for multiple intervals */
/******************************************************/

data dstar;
set step1a;
y_star=B_0hat+B_1hat*x+time_error_star+location_error_star;
good_b1hatstar=0; /* again for generate macro below */
run;



proc datasets library=work;
delete step1a;
run;


/* This finishes step2 */

/****************************************************************/
/** STEP 3
/*
/****************************************************************/



DM "Log; Clear; output; clear";
run;

ods exclude all;

proc mixed data=dstar CL method=REML;
by rep exp;
class t j;
model y_star=x  / cl ddfm=kr solution CovB;
random t;
ods output covparms=var_params_star CovB=Cov_fixstar
solutionf=solution_fixed_star tests3=type3test;
run;

ods exclude none;



data type3test;
set type3test;
good_b1hatstar=0;
if probf<=.05 then good_b1hatstar=1;
run;


/* Now just grab parameter estimates like last time, just by using */


data solution_fixed_star;
set solution_fixed_star;
if effect="Intercept" then effect="B_0hatstar";
if effect="x" then effect="B_1hatstar";
```

```
run;

data var_params_star;
set var_params_star;
if covparm="t" then covparm="time_vhatstar";
if covparm="Residual" then covparm="loc_vhatstar";
run;





proc transpose data=solution_fixed_star out=solution_fixed_star2;
by rep exp;
ID Effect;
run;

proc transpose data=var_params_star out=var_params_star2;
by rep exp;
ID covparm;
run;

proc sql;
create table step3a as
select a.*,
         c.B_0hatsta as B_0hatstar,
         c.B_1hatsta as B_1hatstar,
         d.time_vhat as time_vhatstar,
         d.loc_vhats as loc_vhatstar,
         e.good_b1hatstar as testing_b1hatstar
from dstar as a, solution_fixed_star2 as c,var_params_star2 as d,
type3test as e
where a.exp=c.exp AND a.exp=d.exp AND a.exp=e.exp
                           AND
        a.rep=c.rep AND a.rep=d.rep AND a.rep=e.rep
AND c._name_="Estimate"
AND d._name_="Estimate";
quit;

data step3a;
set step3a;
testing_b1hat=good_b1hat;
drop good_b1hat;
run;




/* Must generate the x_skhats
*/
data step3temp;
do rep=1 to &num_reps;
      do exp=1 to &num_stat;
      x_skhat=0;
      e_skhat_source = rannor(&seed+9);
      time_error_source=rannor(&seed+9);
```

```
      loc_error_source=rannor(&seed+9);
       /* These should be in step 4 but it runs easier */
            do t=1 to &num_t;
                  do j=1 to &num_l;
                        output;
                  END;
            END;
      END;
END;
run;

proc sql;
create table step3b as
select a.*,
         b.x_skhat,
         b.e_skhat_source,
         b.time_error_source,
         b.loc_error_source
from step3temp as b,
       step3a as a
where
a.rep=b.rep AND
a.exp=b.exp AND
a.t=b.t AND
a.j=b.j;
quit;


data step3;
set step3b;
if testing_b1hat = 1 then
    x_skhat=(y_sk-B_0hat-
(sqrt(time_v)*e_skhat_source+sqrt(loc_v)*e_skhat_source))/B_1hat;
else
      x_skhat = .;
run;

proc datasets library=work;
delete step3temp step3a step3b dstar;
run;


/**********/
/* Step 4 */
/**********/
/*
For naming conventions I use the _sk to denote a subscript of sk, the
star suffixed at the end means a stared notation variable, thus:
y_skstar = a variable y with sk subscript that is superscripted with
star.
While this sounds complex, it clearly explains the intended usage of
this variable in our work.*/

/* Generate the error_skstar variables */

data step4;
set step3;
```

```
timeerror_skstar=sqrt(time_v)*time_error_source;
locerror_skstar=sqrt(loc_v)*loc_error_source;
y_skstar=b_0hat+b_1hat*x_skhat+timeerror_skstar+locerror_skstar;
drop time_error_source loc_error_source;
run;



proc datasets library=work;
delete step3;
run;
quit;



/*************************************/
/* End of step 4                     */
/*************************************/

/*** Begin step 5 ***/


proc sql;
create table step5a as
select
    exp,
      rep,
      col1 as var_b0hatstar,
      col2 as cov_b1b0hatstar
from cov_fixstar
where row=1
order by rep, exp;
quit;

proc sql;
create table step5b as
select
    exp,
      rep,
      col2 as var_b1hatstar
from cov_fixstar
where row=2
order by rep, exp;
quit;

proc sql;
create table step5c as
select a.*,
         b.*
from step5a as a, step5b as b
where a.exp=b.exp and a.rep=b.rep;
quit;

proc sql;
create table step5d as
```

```
select a.*,
         b.*
from step5c as a, step4 as b
where
a.exp=b.exp
and
a.rep=b.rep;
run;

data step5;
set step5d;
num_t=&num_t;
num_l=&num_l;
scale = (num_t*num_l) / ( (num_t*num_l) - 2 ) ;
top=y_skstar-B_0hatstar-(B_1hatstar*x_skhat);
bottom=sqrt(time_vhatstar+loc_vhatstar+var_b0hatstar+((x_skhat*x_skhat)
*var_b1hatstar)+2*(x_skhat*cov_b1b0hatstar));
F_star=(scale)*((top/bottom)**2);
drop num_t num_l scale;
run;

proc sql;
create table storage.counting_mi_&ID as
select distinct
exp,
rep,
testing_b1hat,
testing_b1hatstar
from
step5;
quit;


proc datasets library=work;
delete step5a step5b step5c step5d step4;
quit;


/**** Step 6 is already completed. ****/

/* step7 */
proc sort data=step5 out=step7 NODUPKEY;
where testing_b1hat=1 and testing_b1hatstar=1;
by rep exp f_star;
run;


proc univariate data=step7 noprint;
```

```
by rep;
   var F_star;
   output out=percentiles pctlpts=95 pctlpre=P;
run;



/****************** STEP 8 *************************/
/* Computing the confidence intervals                    */
/**************************************************/

data step8a;
set percentiles;
f1=p95;
run;

/* First grab the data needed from tables lying about */
proc sql;
create table step8b as
select distinct
      c.f1,
      a.b_1hat,
      a.b_0hat,
      a.testing_b1hat,
      b.var_b1hat,
      b.var_b0hat,
      a.y_sk,
      b.cov_b1b0hat,
      a.time_v,
      a.loc_v,
      a.x_sk,
      a.rep
from  step7 as a,
            data_prep8c as b,
            step8a as c
where
      c.rep=a.rep AND
      c.rep=b.rep AND
      testing_b1hat=1
order by rep;
quit;


data quadratic_coefficients;
set step8b;
a = (b_1hat**2)-(var_b1hat*f1);
b = ((b_0hat*b_1hat)-(b_1hat*y_sk)-(cov_b1b0hat*f1));
c = (y_sk**2)-2*(b_0hat*y_sk)+(b_0hat**2)-f1*(time_v+loc_v+var_b0hat);
run;

data storage.quadratic_coefficients_&ID;
      set quadratic_coefficients;
run;

proc sort data=quadratic_coefficients;
by a;
```

```
run;

/* finally the confidence interval */
data step8(KEEP=a b c lower upper rep x_sk p_score);
set quadratic_coefficients;
lower=(-2*b-sqrt((4*b**2)-4*a*c))/(2*a);
upper=(-2*b+sqrt((4*b**2)-4*a*c))/(2*a);
if (x_sk>=lower) AND (x_sk<= upper) AND (lower^=.) AND (upper^=.) then
p_score=1;
else if lower=. AND upper=. then p_score=-1;
else p_score=0;
run;



proc sql;
create table p_valinformation as
select count(*) as total,
       sum(p_score) as successes
from step8 where
p_score^=-1;
quit;

data p_valinformation;
 set p_valinformation;
 coverage=successes/total;
run;




/* At the end of the simulation now need to count up proper scores
      1. Count the number of times proc mixed grabbed the true value of
B_1hat B_0hat and
         time error and location error hats.
      2. Same for Stars.
      3. Coverage rate for x_sk
      4. Mean length of intervals
      5. Number of runs actually computed.

 */

/* For this table our rates of estimation only care about where B_1hat
was estimated to not be 0.
*/

/* This is used as a reference table for calculations */
/* First part is to tally the parameter estimates */
/* Begin with only checking those experiments that have B_1hat being
non-zero.
   This is done by establishing a reference table that tells you what
the good real
   experiments are. Then cross referencing that with other tables */
proc sql;
create table
good_reference as
select distinct
```

```
      testing_b1hat,
      rep
from step5
where
      testing_b1hat=1;
quit;

proc sql;
create table b0b1_checking as
select a.*
from solution_fixed as a, good_reference as b
where
      a.rep=b.rep;
quit;

proc sql;
create table time_loc_varchecking as
select a.*
from var_params as a, good_reference as b
where
      a.rep=b.rep;
quit;

data b0b1_checking;
      set b0b1_checking;

      intercept_score=0;
      if &intercept>= lower AND &intercept<=upper then
intercept_score=1;

      slope_score=0;
      if &slope>= lower and &slope<=upper then slope_score=1;
run;

data time_loc_varchecking;
      set time_loc_varchecking;

      time_score=0;
      if &var_terror >= lower AND &var_terror <= upper then
time_score=1;

      loc_score=0;
      if &var_lerror >= lower AND &var_lerror <= upper then
loc_score=1;
run;

proc summary data=b0b1_checking;
class effect;
var intercept_score slope_score;
output out=b0b1_score sum=;
run;

proc summary data=time_loc_varchecking;
class covparm;
var time_score loc_score;
output out=timeloc_score sum=;
run;
```

```
/* Now to do the same with the star sets */
/* Must form a reference table again, I only want to check those that
   had B_1hat being not 0, and B_1hatstar being not 0
 */

proc sql;
create table good_reference_star as
select
      distinct
      a.rep,
      a.exp,
      a.testing_b1hat,
      a.testing_b1hatstar
      from
      step5 as a
where
 a.testing_b1hat=1 AND
 a.testing_b1hatstar=1;
quit;


proc sql;
create table b0b1star_checking as
select a.*
from solution_fixed_star as a, good_reference_star as b
where
      a.rep=b.rep AND
      a.exp=b.exp;
quit;

proc sql;
create table timestar_locstar_varchecking as
select a.*
from var_params_star as a, good_reference_star as b
where
      a.rep=b.rep AND
      a.exp=b.exp;
quit;

data b0b1star_checking;
      set b0b1star_checking;

      intercept_score=0;
      if &intercept>= lower  AND &intercept<=upper then
intercept_score=1;

      slope_score=0;
      if &slope>= lower and &slope<=upper then slope_score=1;
run;

data timestar_locstar_varchecking;
      set timestar_locstar_varchecking;

      time_score=0;
      if &var_terror >= lower AND &var_terror <= upper then
time_score=1;
```

```
      loc_score=0;
      if &var_lerror >= lower AND &var_lerror <= upper then
loc_score=1;
run;



proc summary data=b0b1star_checking;
class effect;
var intercept_score slope_score;
output out=b0b1star_score sum=;
run;

proc summary data=timestar_locstar_varchecking;
class covparm;
var time_score loc_score;
output out=timelocstar_score sum=;
run;

/* Now we need to put this data set together */




/* Grab data from the score tables and put it into variables */


proc sql;
select intercept_score into :b0_hatscores
from b0b1_score
where effect="B_0hat";
quit;

proc sql;
select slope_score into :b1_hatscores
from b0b1_score
where effect="B_1hat";
quit;

proc sql;
select time_score into :time_varscores
from timeloc_score
where covparm="time_v";
quit;

proc sql;
select loc_score into :loc_varscores
from timeloc_score
where covparm="loc_v";
quit;

proc sql;
select _freq_ into :good_b1hats
from b0b1_score
where effect="B_1hat";
quit;
```

```
/* Now we adress stars */

proc sql;
select intercept_score into :b0star_hatscores
from b0b1star_score
where effect="B_0hatsta";
quit;

proc sql;
select slope_score into :b1star_hatscores
from b0b1star_score
where effect="B_1hatsta";
quit;

proc sql;
select time_score into :timestar_varscores
from timelocstar_score
where covparm="time_vhat";
quit;

proc sql;
select loc_score into :locstar_varscores
from timelocstar_score
where covparm="loc_vhats";
quit;

proc sql;
select _freq_ into :good_b1hatsandstars
from b0b1star_score
where effect="B_1hatsta";
quit;



/* Now need to do calculations for statistics based on the confidence
intervals:
      1. Get the mean length.
      2. Count the number of good ones
      3. Give the numbers for a coverage rate.
  */

/*  define the length of a CI as abs(upper-lower) */
data step8;
set step8;
length=abs(upper-lower);
run;

/* Now use proc means to obtain the length where the CS is valid:
      p_score=1 or p_score=0
 */

proc summary data=step8;
where p_score=1 or p_score=0;
var length;
output out=CIlengths mean=averagelength;
```

```
run;

/* The data set P_valinformation has all the information about coverage
for 2. */

proc sql;
select averagelength into :average_ci_length
from cilengths;
quit;

proc sql;
select coverage into :ci_coverage
from p_valinformation;
select successes into :ci_goodcount
from p_valinformation;
select total into :total_cis
from p_valinformation;
quit;




/* Final report datasets */
data storage.final_report_&ID;
b0=&b0_hatscores;
b1=&b1_hatscores;
timev=&time_varscores;
locv=&loc_varscores;
b1_hatcount=&good_b1hats;

b0_star=&b0star_hatscores;
b1_star=&b1star_hatscores;
timev_star=&timestar_varscores;
locv_star=&locstar_varscores;
b1_hatstarcount=&good_b1hatsandstars;

Total_possible_b1hats=&num_reps;
Total_possible_b1hatstars=%eval(&num_reps*&num_stat);

average_length=&average_ci_length;
x_skci_successes=&ci_goodcount;
x_skci_total=&total_cis;
coverage=&ci_coverage;

bad_cis=Total_possible_b1hats-x_skci_total;

run;


proc sql;
create table diagnostic_&id as
select distinct
rep,
exp,
testing_b1hat,
testing_b1hatstar
from step5;
```

```
quit;


proc summary data=diagnostic_&id;
class rep;
var testing_b1hat testing_b1hatstar;
output out=sums sum=;
run;

data storage.diagnostic_&id(keep= rep testing_b1hat testing_b1hatstar
rate);
set sums;
if testing_b1hat>0 then testing_b1hat=1;
if rep=. then delete;
rate=testing_b1hatstar/&num_stat;
run;


quit;

%mend;

%do_simulation(
  seed=11099504,
  alpha=.05,
  num_t=8,
  num_l=6,
  intercept=0,
  slope=20,
  var_terror=.05,
  var_lerror=5,
  num_stat=50,
  num_reps=50,
  ID=31110);
run;
```

The following is the code I used to merge simulation results across computers and perform additional analysis of simulations.

```
/*****************************************************************
/* This program merges all results from every computer and puts them
all together into one
 file

There are three datasets for this:
      processed_mis = performance of the bootstrap per confidence
interval per sim.
      methods = counts if x_sk was in one ci vs another type of ci (
since we had two types )
      comparison_table = compares the means and stds across the
methods.



 */

libname storage 'C:\real_case';
run;

%let datasetlist="11110,
11120,
11210,
11220,
12110,
12120,
12210,
12220,
13110,
13120,
13210,
13220,
21110,
21120,
21210,
21220,
22110,
22120,
22210,
22220,
23110,
23120,
23210,
23220,
31110,
31120,
31210,
31220,
32110,
32120,
32210,
```

```
32220,
33110,
33120,
33210,
33220";
run;

%let num_datasets=36;


data merged_results;
run;

%macro combine();

%DO I=1 %to &num_datasets;
 %let current_dataset=%scan(&datasetlist,&I,",");


     proc contents data=storage.final_report_&current_dataset;
       run;

       proc sql;
       create table new_result as
       select
       a.*,
       &current_dataset as ID
       from
       storage.final_report_&current_dataset as a;
       quit;

       data merged_results;
       set merged_results new_result;
       run;


%END;

proc sql;
create table new_results as
select distinct * from merged_results;
quit;

%MEND;

%combine();
run;

proc sort data=new_results;
by id;
run;


/*******************************************************/
/* Now compute the cis using the percentile method       */
/*******************************************************/
```

```
%let datasetlist="11110,
11120,
11210,
11220,
12110,
12120,
12210,
12220,
13110,
13120,
13210,
13220,
21110,
21120,
21210,
21220,
22110,
22120,
22210,
22220,
23110,
23120,
23210,
23220,
31110,
31120,
31210,
31220,
32110,
32120,
32210,
32220,
33110,
33120,
33210,
33220";
run;

%let num_datasets=36;



data results_percentile;
run;



%macro average_sd_percentile();

%DO I=1 %to &num_datasets;
 %let current_dataset=%scan(&datasetlist,&I,",");

    data ci_formula;
    set storage.Quadratic_coefficients_&current_dataset;
    lower  = (-2*b - sqrt((4*b**2)-4*a*c) )/(2*a);
    upper = (-2*b + sqrt((4*b**2)-4*a*c) )/(2*a);
    distance = abs(upper-lower);
```

```
      if (x_sk>=lower) AND (x_sk<= upper) AND (lower^=.) AND (upper^=.)
then p_score=1;
    else if lower=. AND upper=. then p_score=-1;
    else p_score=0;
    run;

      data ci_&current_dataset;
       set ci_formula;
      run;

      proc freq data=ci_formula;
      where p_score in (1,0);
      tables p_score / out=computing_coverage;
      run;

      proc sql;
      select percent into :coverage
      from computing_coverage
      where p_score=1;
    quit;


    proc means data=ci_formula;
      where p_score=1 or p_score=0;
    var distance;
    output out=ci_means_sds;
    run;

    proc transpose data=ci_means_sds out=flipped_ci_sds;
    id _stat_;
    run;

    proc sql;
    create table atom as
    select
    "&current_dataset" as id,
    mean,
    std,
      &coverage as coverage
    from flipped_ci_sds
    where _NAME_="distance";
    quit;

      data results_percentile;
       set results_percentile atom;
      run;

%END;

%MEND;

%average_sd_percentile();
run;

data ci_22220_percentile;
 set ci_22220;
run;
```

```
/*****************************************/
/* Newer versions of the percentiles
   using an F distribution
 */

data results_fconf;
run;

data methods;
run;


data loctime_keys;
      set storage.time_locations_table;
      if id=. then delete;
run;


%let datasetlist="11110,
11120,
11210,
11220,
12110,
12120,
12210,
12220,
13110,
13120,
13210,
13220,
21110,
21120,
21210,
21220,
22110,
22120,
22210,
22220,
23110,
23120,
23210,
23220,
31110,
31120,
31210,
31220,
32110,
32120,
32210,
32220,
33110,
33120,
33210,
33220";
```

```
run;

%let num_datasets=36;



%macro average_sd_tformula();

%DO I=1 %to &num_datasets;
 %let current_dataset=%scan(&datasetlist,&I,",");

      /* Need to grab the number of locations and times into the new
f_statistic */

      proc sql;
      select num_t into :num_times from loctime_keys
      where id=&current_dataset;
      quit;

      proc sql;
      select num_l into :num_locs from loctime_keys
      where id=&current_dataset;
      quit;

      %let bottom_df=%eval(&num_times*&num_locs);
      run;


    data ci_formula;
    set storage.Quadratic_coefficients_&current_dataset;
      f_new=finv(.95,1,&bottom_df-2);
      af = (b_1hat**2)-(var_b1hat*f_new);
    bf = ((b_0hat*b_1hat)-(b_1hat*y_sk)-(cov_b1b0hat*f_new));
    cf = (y_sk**2)-2*(b_0hat*y_sk)+(b_0hat**2)-
f_new*(time_v+loc_v+var_b0hat);
    lower  = (-2*bf - sqrt((4*bf**2)-4*af*cf) )/(2*af);
    upper = (-2*bf + sqrt((4*bf**2)-4*af*cf) )/(2*af);
    distance = abs(upper-lower);
      if (x_sk>=lower) AND (x_sk<= upper) AND (lower^=.) AND (upper^=.)
then p_score=1;
    else if lower=. AND upper=. then p_score=-1;
    else p_score=0;

      lower_p  = (-2*b - sqrt((4*b**2)-4*a*c) )/(2*a);
    upper_p = (-2*b + sqrt((4*b**2)-4*a*c) )/(2*a);
    distance_p = abs(upper_p-lower_p);
      if (x_sk>=lower_p) AND (x_sk<= upper_p) AND (lower_p^=.) AND
(upper_p^=.) then pp_score=1;
    else if lower_p=. AND upper_p=. then pp_score=-1;
    else pp_score=0;
    run;

      data ci_&current_dataset;
       set ci_formula;
      run;

      proc freq data=ci_formula;
```

```
  where p_score in (1,0);
  tables p_score / out=computing_coverage;
  run;

  proc freq data=ci_formula;
  tables p_score / out=inves_p_&current_dataset;
  run;

  proc freq data=ci_formula;
  where ( p_score in (1,0) ) AND ( pp_score in (1,0) );
  tables p_score*pp_score / SPARSE out=compare_table;
  run;

  data compare_table;
  set compare_table;
  id=&current_dataset;
  run;

  proc sql;
  create table method_&current_dataset as
  select
  id,
  p_score as in_f_approach,
  pp_score as in_perc_approach,
  count,
  percent
  from
  compare_table;
  quit;


  data methods;
        set methods method_&current_dataset;
  run;

  proc sql;
  select percent into :coverage
  from computing_coverage
  where p_score=1;
quit;


proc means data=ci_formula;
  where p_score=1 or p_score=0;
var distance;
output out=ci_means_sds;
run;

proc transpose data=ci_means_sds out=flipped_ci_sds;
id _stat_;
run;

proc sql;
create table atom as
select
"&current_dataset" as id,
mean,
```

```
    std,
      &coverage as coverage
    from flipped_ci_sds
    where _NAME_="distance";
    quit;

      data results_fconf;
       set results_fconf atom;
      run;

%END;

%MEND;

%average_sd_tformula();
run;



/******************/
/* Now do the mi's */
/******************/




data processed_mi;
run;


%macro do_mis();


%DO I=1 %to &num_datasets;
 %let current_dataset=%scan(&datasetlist,&I,",");


proc sort data=storage.counting_mi_&current_dataset;
by rep;
run;

proc summary data=storage.counting_mi_&current_dataset;
by rep;
var testing_b1hatstar;
output out=tallies sum=tally_perrep;
run;

proc means data=tallies;
var tally_perrep;
output out=results_tallies;
run;

proc transpose data=results_tallies out=temp;
ID _STAT_;
run;

proc sql;
```

```
create table temp as
select
&current_dataset as id,
mean as mean_mis,
std as std_mis
from
temp
where
_NAME_="tally_perrep";
quit;



data processed_mi;
      set processed_mi temp;
run;

%END;

%MEND;

%do_mis();
run;

/* FINALLY, we simply want to merge the result datasets about coverages
and mean, sd together */
proc sql;
create table comparison_table as
select
a.id,
a.mean as mean_f,
a.std as std_f,
a.coverage as coverage_f,
b.mean as mean_perc,
b.std as std_perc,
b.coverage as coverage_perc
from
results_fconf as a
 left join
results_percentile as b
 on
a.id=b.id;
quit;

/***************************/
/* learning proc tabulate */
/***************************/



%let datasetlist="11110,
11120,
11210,
11220,
12110,
12120,
12210,
```

```
12220,
13110,
13120,
13210,
13220,
21110,
21120,
21210,
21220,
22110,
22120,
22210,
22220,
23110,
23120,
23210,
23220,
31110,
31120,
31210,
31220,
32110,
32120,
32210,
32220,
33110,
33120,
33210,
33220";
run;

%let num_datasets=36;

%macro make_tables();


ODS HTML FILE="C:\TEMP.XLS";


%DO I=1 %to &num_datasets;
 %let num=%scan(&datasetlist,&I,",");

  data ci_&num;
      set ci_&num;
      if p_score = 1 then Method_1 = "Yes";
      else if p_score= 0 then method_1 = "No";
      if pp_score= 1 then Method_2 = "Yes";
      else if pp_score=0 then method_2 = "No";
  run;

  proc freq order=data data=ci_&num;
   title1 "Count of Method 1 vs. Method 2 for sim: &num";
      tables Method_1*Method_2 / agree NOPERCENT NOROW NOCOL;
  run;

%END;
```

```
ODS HTML CLOSE;
run;

%mend();

%make_tables();
run;
```

Appendix B- Simulation Tables

The following tables were made to summarize the results of our simulation study.

Table B.1 Table of Mean Interval Length and Standard Deviation

Simulation ID	Method 1 Coverage	Method 1 Mean Length	Method 1 Standard Deviation of Length	Method 2 Coverage	Method 2 Mean Length	Method 2 Standard Deviation of Length
11110	96.51	9.00	15.73	89.29	10.19	29.28
11120	94.29	6.63	3.72	92.57	6.46	4.48
11210	95.04	15.41	87.89	93.97	8.57	20.91
11220	94.90	6.74	8.73	93.33	6.43	5.21
12110	91.94	18.60	43.34	89.83	16.44	49.29
12120	92.09	14.99	33.81	91.43	11.94	11.32
12210	93.40	14.00	18.90	91.09	18.54	60.30
12220	91.62	8.54	4.21	92.15	9.66	13.40
13110	91.00	4.64	1.21	94.00	5.56	1.60
13120	90.00	4.54	1.22	92.00	5.43	1.58
13210	90.00	4.61	1.00	92.00	5.15	1.24
13220	92.50	4.63	2.88	94.00	5.16	3.10
21110	96.50	1.34	0.63	95.50	1.37	0.61
21120	96.00	1.64	0.28	96.00	1.72	0.36
21210	95.50	1.70	0.36	95.50	1.78	0.42
21220	92.00	1.61	0.21	93.00	1.68	0.30
22110	92.50	1.81	0.46	94.00	1.94	0.55
22120	92.50	1.61	0.27	94.50	1.69	0.35
22210	97.00	1.71	0.35	97.00	1.80	0.44
22220	95.00	1.58	0.19	95.00	1.64	0.26
23110	91.00	1.18	0.32	96.00	1.41	0.40
23120	91.00	1.10	0.31	94.50	1.32	0.42
23210	93.47	1.16	0.23	94.47	1.30	0.29
23220	92.00	1.18	0.75	92.50	1.31	0.82
31110	94.00	0.46	0.05	93.50	0.47	0.07
31120	94.00	0.45	0.03	94.00	0.45	0.04
31210	94.50	0.46	0.05	94.50	0.47	0.06
31220	96.00	0.44	0.02	97.00	0.45	0.04
32110	93.50	0.67	0.12	93.50	0.71	0.16
32120	96.00	0.64	0.09	97.00	0.67	0.12
32210	93.00	0.65	0.10	94.00	0.68	0.12
32220	92.50	0.64	0.08	93.00	0.67	0.10
33110	91.00	0.46	0.13	95.00	0.54	0.16
33120	90.00	0.44	0.12	95.50	0.53	0.15
33210	92.50	0.45	0.10	95.00	0.50	0.12
33220	91.00	0.45	0.10	95.50	0.50	0.12

Appendix C- R Code Used to Make Figures of Results

The following code produces the graphs within my work. Please note that one must load the R package called 'gplots' in order to generate the whisker plots produced by this code.

```
##############################
# for 'method 2'#
# BY SLOPE #
############

x<-read.csv("C:\\celeste\\simulation_c\\MakeGraphs\\graph_by_error_data.csv")
y<-1:36

maintitle=""
ylabval="Mean Length"
par(xaxt="n")
plotCI(y[1:12],x$MEAN[1:12],uiw=1.96*x$STD[1:12]*(1/sqrt(x$n[1:12])),main=mainti
tle,xlab="Simulations where Slope = 2",ylab=ylabval,gap=0,ylim=c(2,35))
par(xaxt="n")
plotCI(y[13:24],x$MEAN[13:24],uiw=1.96*x$STD[13:24]*(1/sqrt(x$n[13:24])),main=
maintitle,xlab="Simulations where Slope = 8",ylab=ylabval,gap=0,ylim=c(1,2.4))
par(xaxt="n")
plotCI(y[25:36],x$MEAN[25:36],uiw=1.96*x$STD[25:36]*(1/sqrt(x$n[25:36])),main=
maintitle,xlab="Simulations where Slope = 20",ylab=ylabval,gap=0,ylim=c(0.3,0.8))

##############################
# for 'method 1'#
# BY SLOPE                   #
```

```
############################

x<-read.csv("C:\\celeste\\simulation_c\\MakeGraphs\\f_graph_by_error_data2.csv")
y<-1:36

maintitle=""
ylabval="Mean Length"

par(xaxt="n")
plotCI(y[1:12],x$MEAN[1:12],uiw=1.96*x$STD[1:12]*(1/sqrt(x$nf[13:24])),main=mai
ntitle,xlab="Simulations where Slope = 2",ylab=ylabval,gap=0,ylim=c(2,35))
par(xaxt="n")
plotCI(y[13:24],x$MEAN[13:24],uiw=1.96*x$STD[13:24]*(1/sqrt(x$nf[13:24])),main=
maintitle,xlab="Simulations where Slope = 8",ylab=ylabval,gap=0,ylim=c(1,2.4))
par(xaxt="n")
plotCI(y[25:36],x$MEAN[25:36],uiw=1.96*x$STD[25:36]*(1/sqrt(x$nf[13:24])),main=
maintitle,xlab="Simulations where Slope = 20",ylab=ylabval,gap=0,ylim=c(0.3,0.8))

##THE FOLLOWING CODE CREATES THE CONFIDENCE INTERVALS
WHISKERS #
############################ by VARIANCE ratio ###########################
#for ‘method 2’#
x<-read.csv("C:\\celeste\\simulation_c\\MakeGraphs\\graph_by_error_data_varc.csv")
y<-1:36

maintitle=""
ylabval="Mean Length"
par(xaxt="n")
plotCI(y[1:12],x$MEAN[1:12],uiw=1.96*x$STD[1:12]*(1/sqrt(x$n[1:12])),main=mainti
tle,xlab="Simulations where Variance Ratio = .05/5",ylab=ylabval,gap=0,ylim=c(0,16))
par(xaxt="n")
```

```
plotCI(y[13:24],x$MEAN[13:24],uiw=1.96*x$STD[13:24]*(1/sqrt(x$n[13:24])),main=
maintitle,xlab="Simulations where Variance Ratio =
5/5",ylab=ylabval,gap=0,ylim=c(0,32))
par(xaxt="n")
plotCI(y[25:36],x$MEAN[25:36],uiw=1.96*x$STD[25:36]*(1/sqrt(x$n[25:36])),main=
maintitle,xlab="Simulations where Variance Ratio =
5/.05",ylab=ylabval,gap=0,ylim=c(0,6))

#############################
# for 'method 1' #
#############################

x<-read.csv("C:\\celeste\\simulation_c\\MakeGraphs\\f_graph_by_error_data3.csv")
y<-1:36

maintitle=""
ylabval="Mean Length"

par(xaxt="n")
plotCI(y[1:12],x$MEAN[1:12],uiw=1.96*x$STD[1:12]*(1/sqrt(x$nf[13:24])),main=mai
ntitle,xlab="Simulations where Variance Ratio =
.05/5",ylab=ylabval,gap=0,ylim=c(0,16))
par(xaxt="n")
plotCI(y[13:24],x$MEAN[13:24],uiw=1.96*x$STD[13:24]*(1/sqrt(x$nf[13:24])),main=
maintitle,xlab="Simulations where Variance Ratio =
5/5",ylab=ylabval,gap=0,ylim=c(0,32))
par(xaxt="n")
plotCI(y[25:36],x$MEAN[25:36],uiw=1.96*x$STD[25:36]*(1/sqrt(x$nf[13:24])),main=
maintitle,xlab="Simulations where Variance Ratio =
5/.05",ylab=ylabval,gap=0,ylim=c(0,6))
```

Appendix D- Figures Discussed in Section 4.2

Figure D.1: Method 1: 95% confidence intervals for mean length when β=2

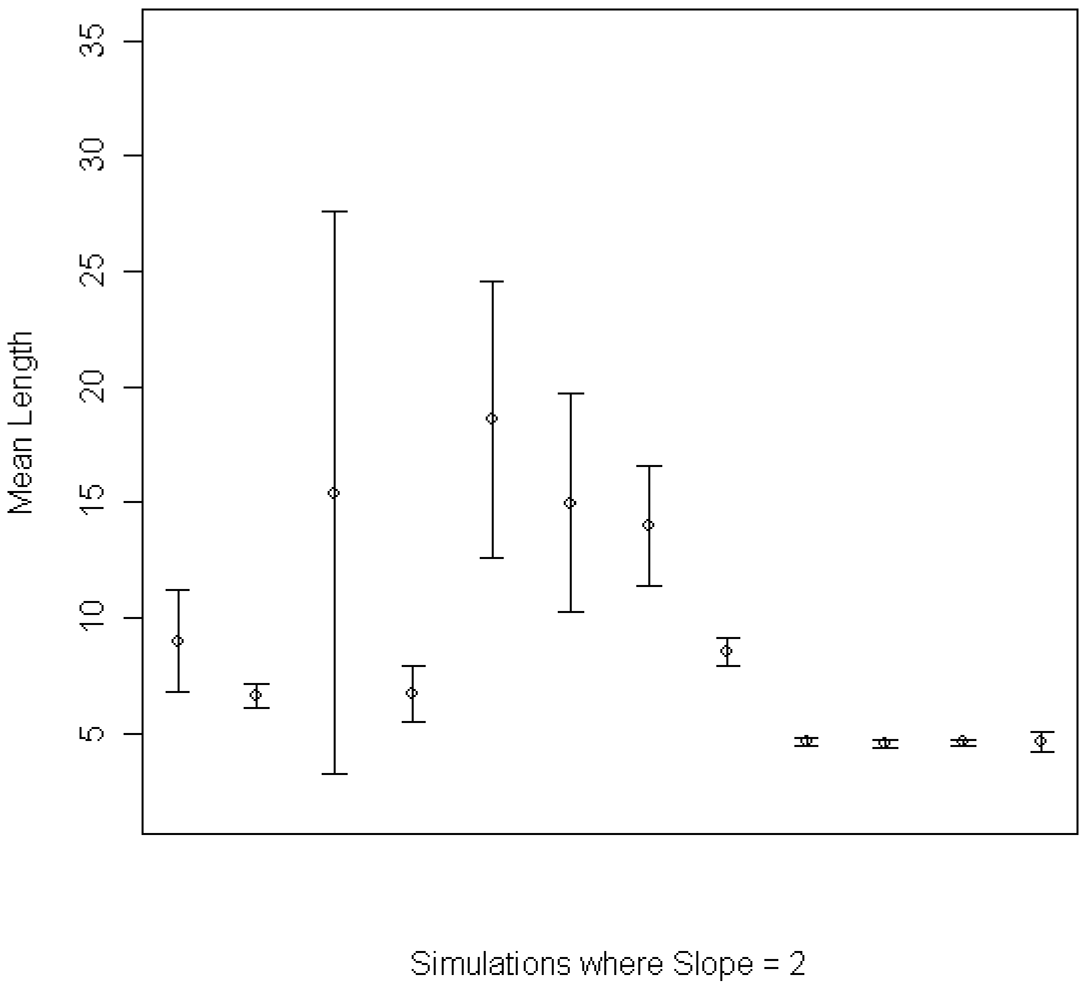


Figure D.2: Method 2: 95% confidence intervals for mean length when β=2

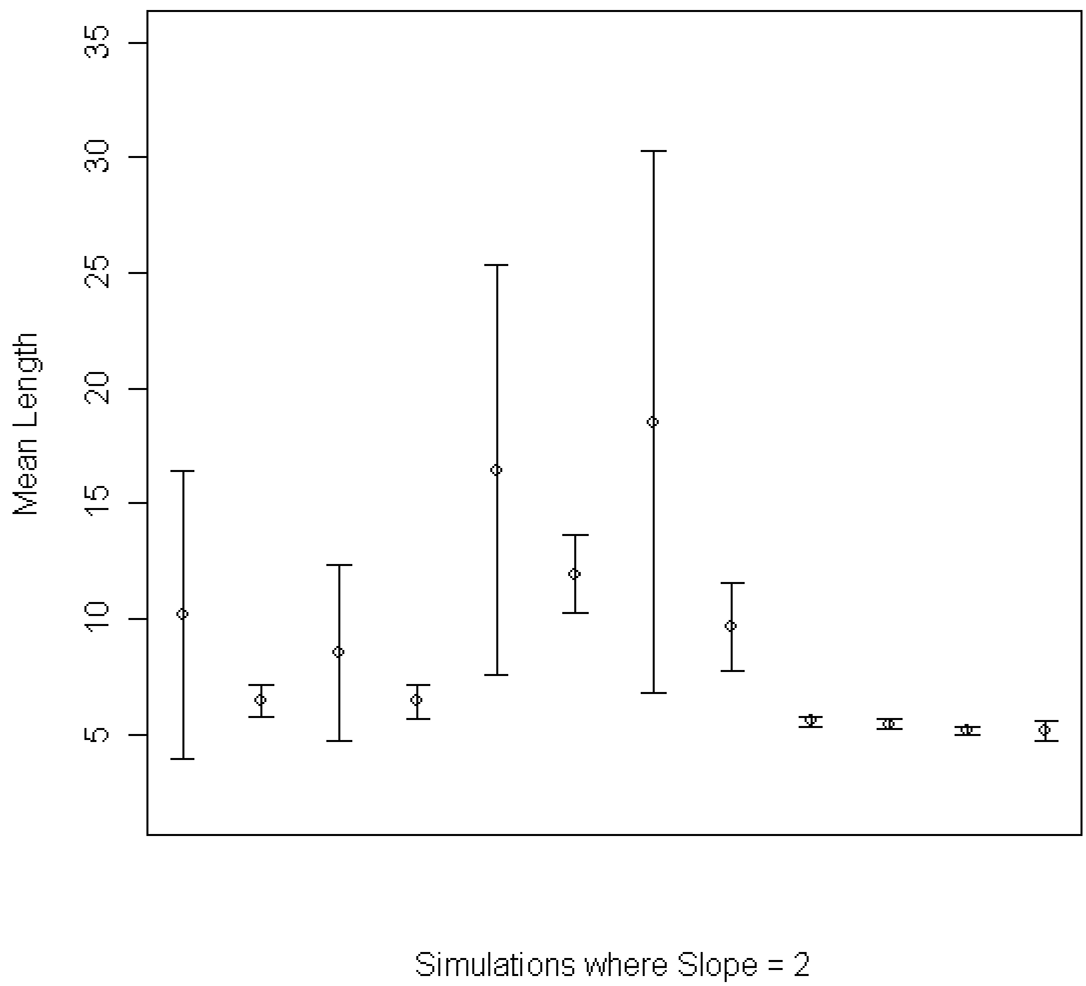


Figure D.3: Method 1: 95% confidence intervals for mean length when β=8

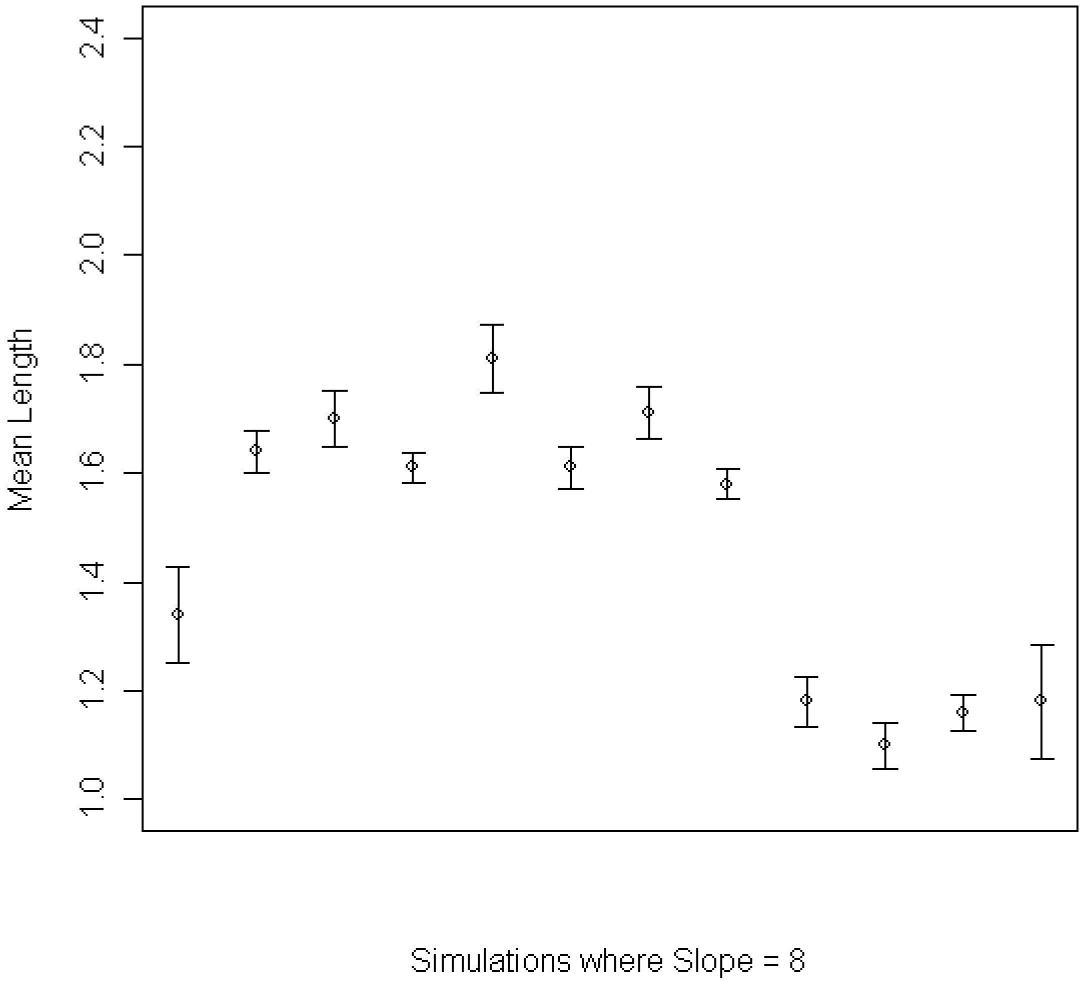


Figure D.4: Method 2: 95% confidence intervals for mean length when β=8

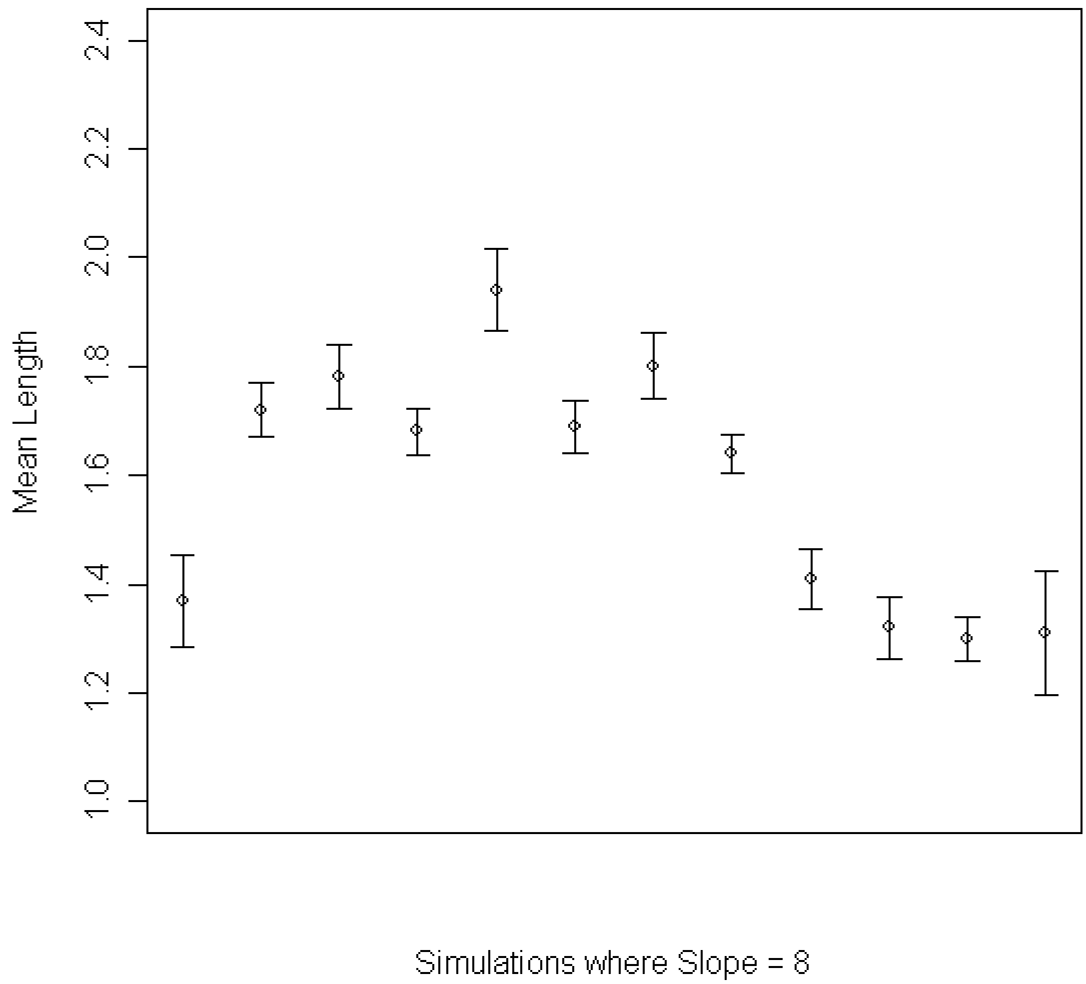


Figure D.5: Method 1: 95% confidence intervals for mean length when β=20

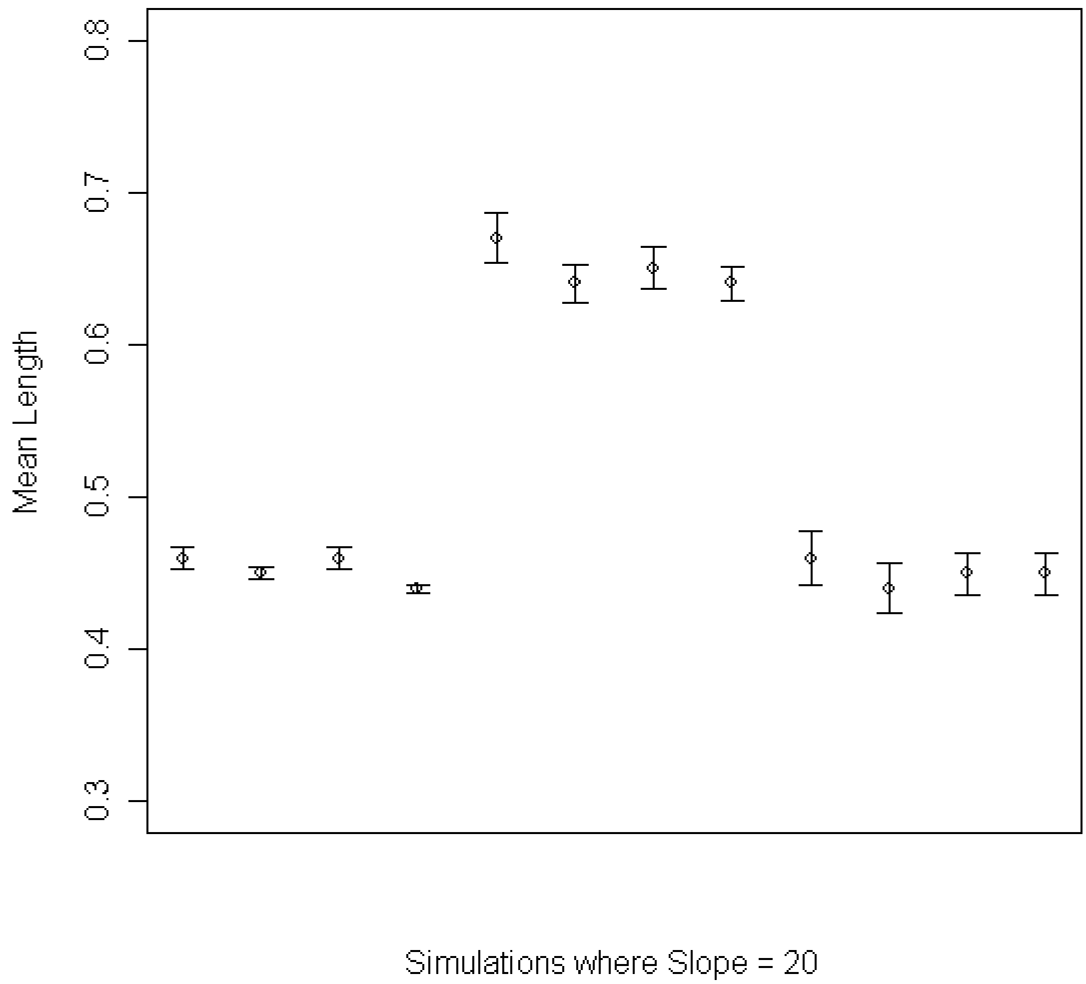


Figure D.6: Method 2: 95% confidence intervals for mean length when β=20

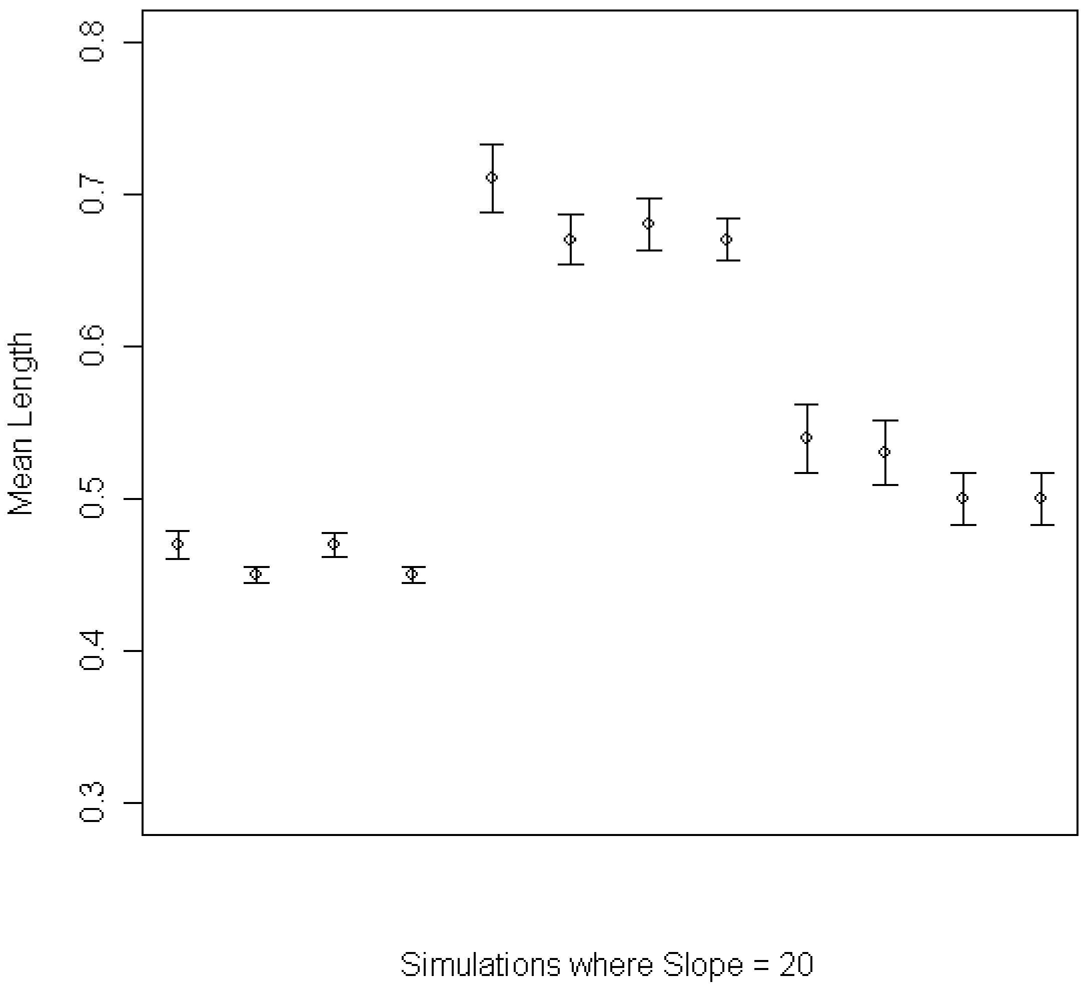


Figure D.7: Method 1: 95% confidence intervals for mean length when $\sigma_{\eta} / \sigma_{\varepsilon}$=.05/5

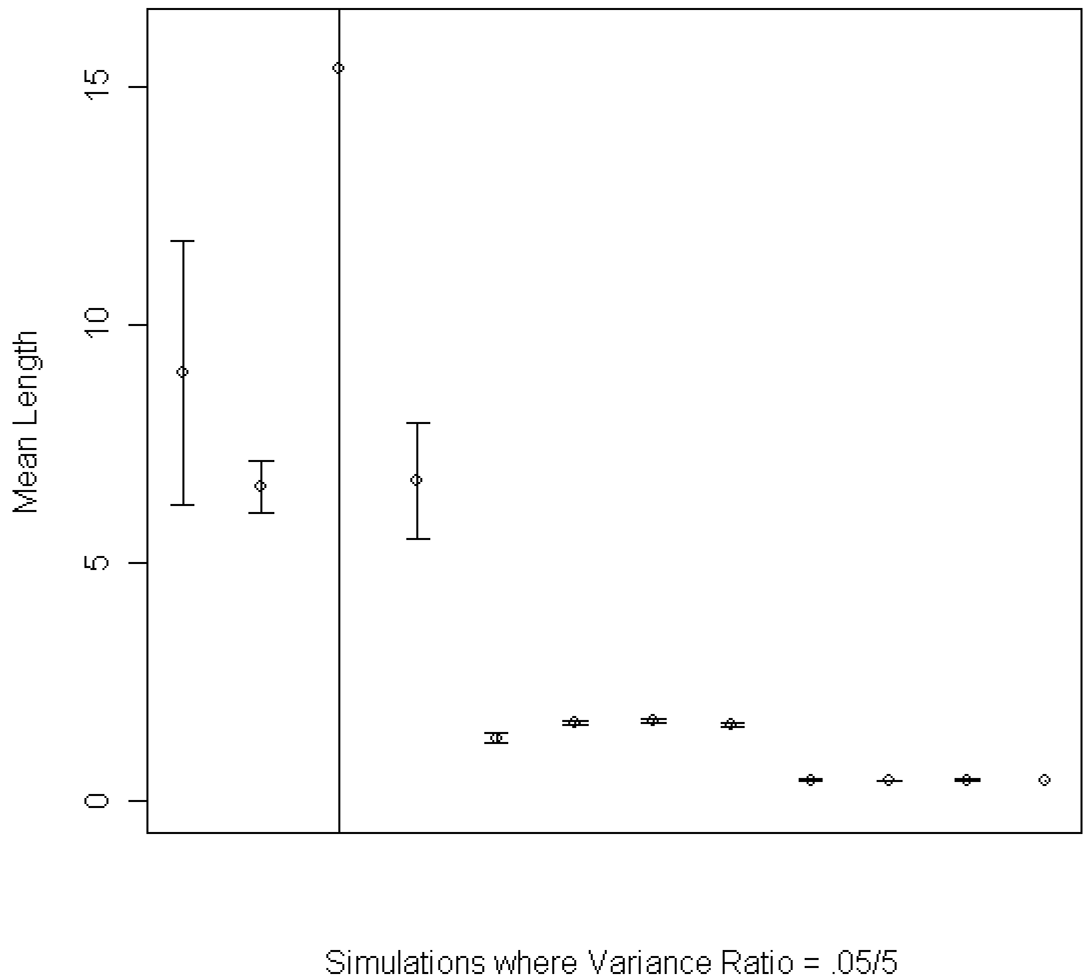


Figure D.8: Method 2: 95% confidence intervals for mean length when $\sigma_\eta / \sigma_\varepsilon$=.05/5

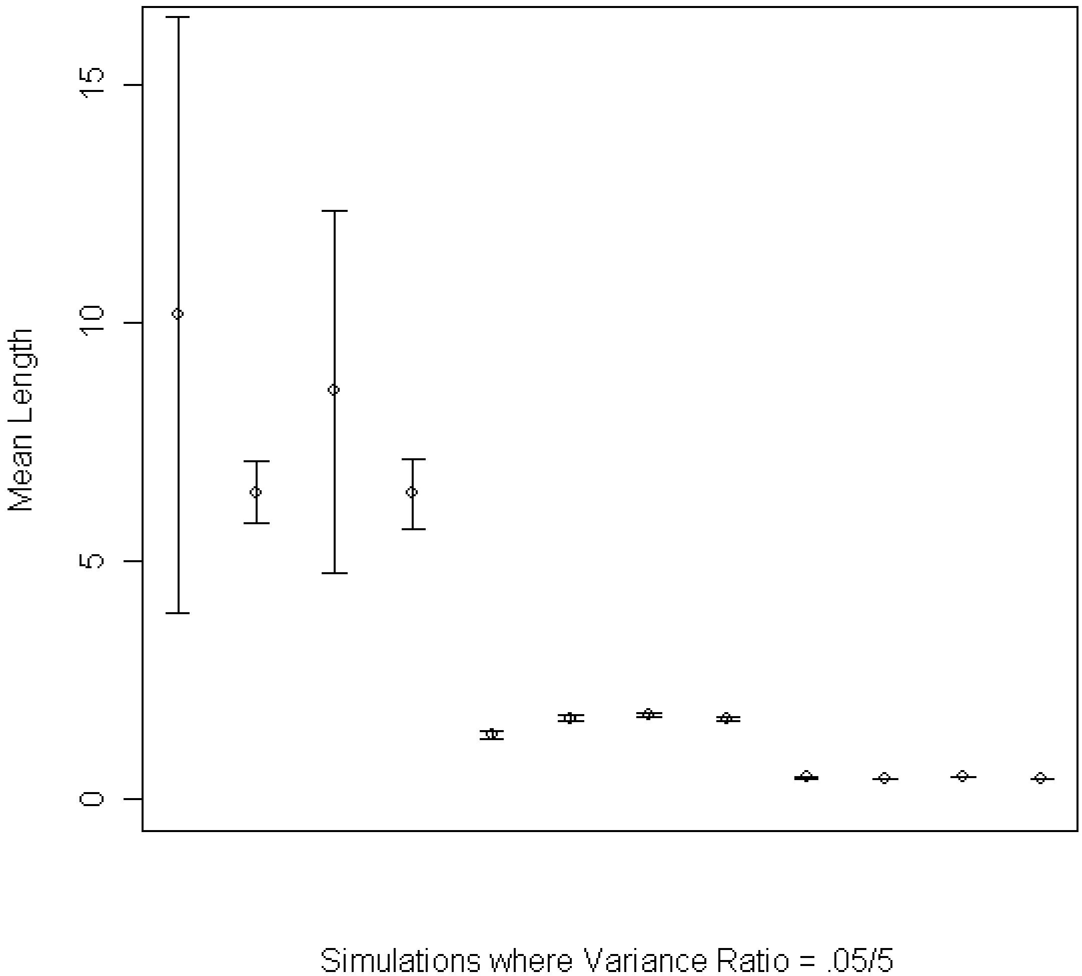


Figure D.9: Method 1: 95% confidence intervals for mean length when $\sigma_\eta / \sigma_\varepsilon$=5/5

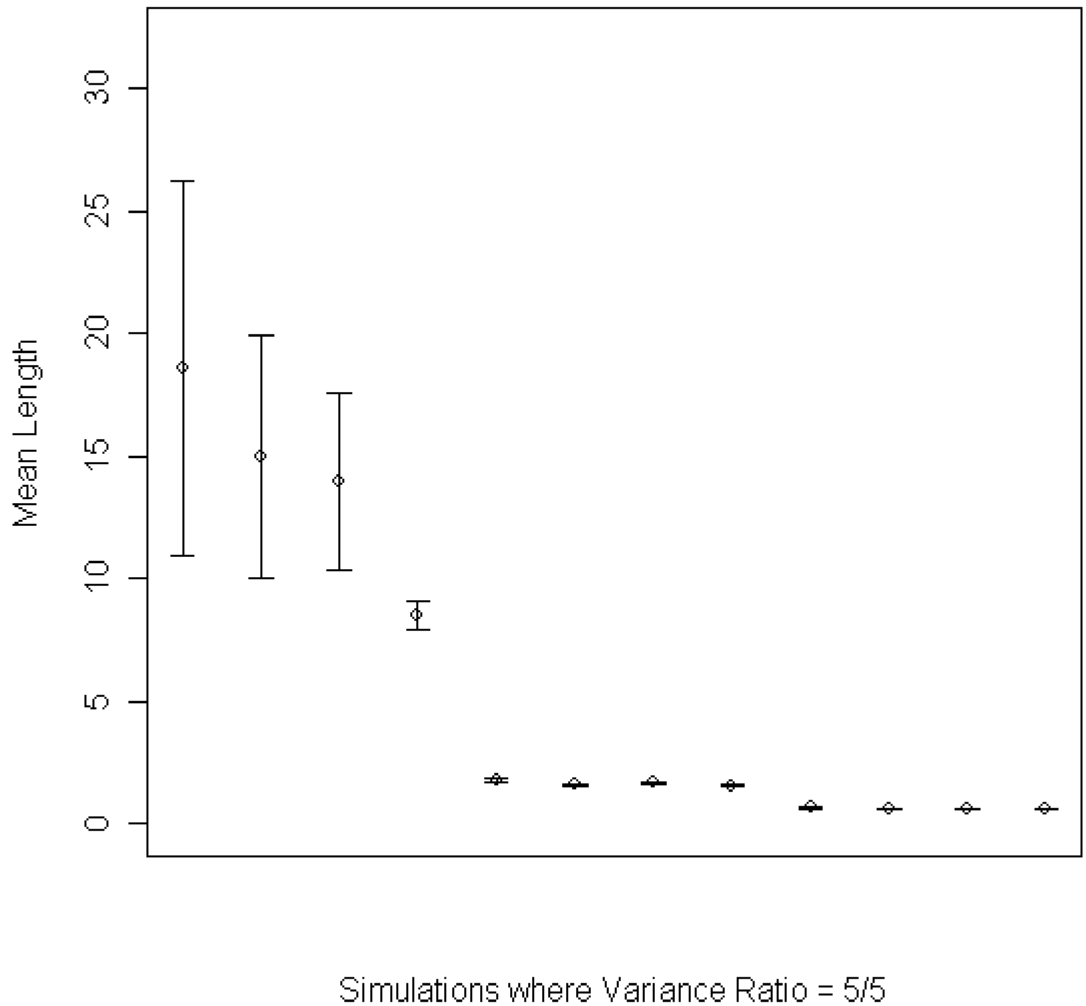


Figure D.10: Method 2: 95% confidence intervals for mean length when $\sigma_\eta^2 / \sigma_\varepsilon^2 = 5/5$

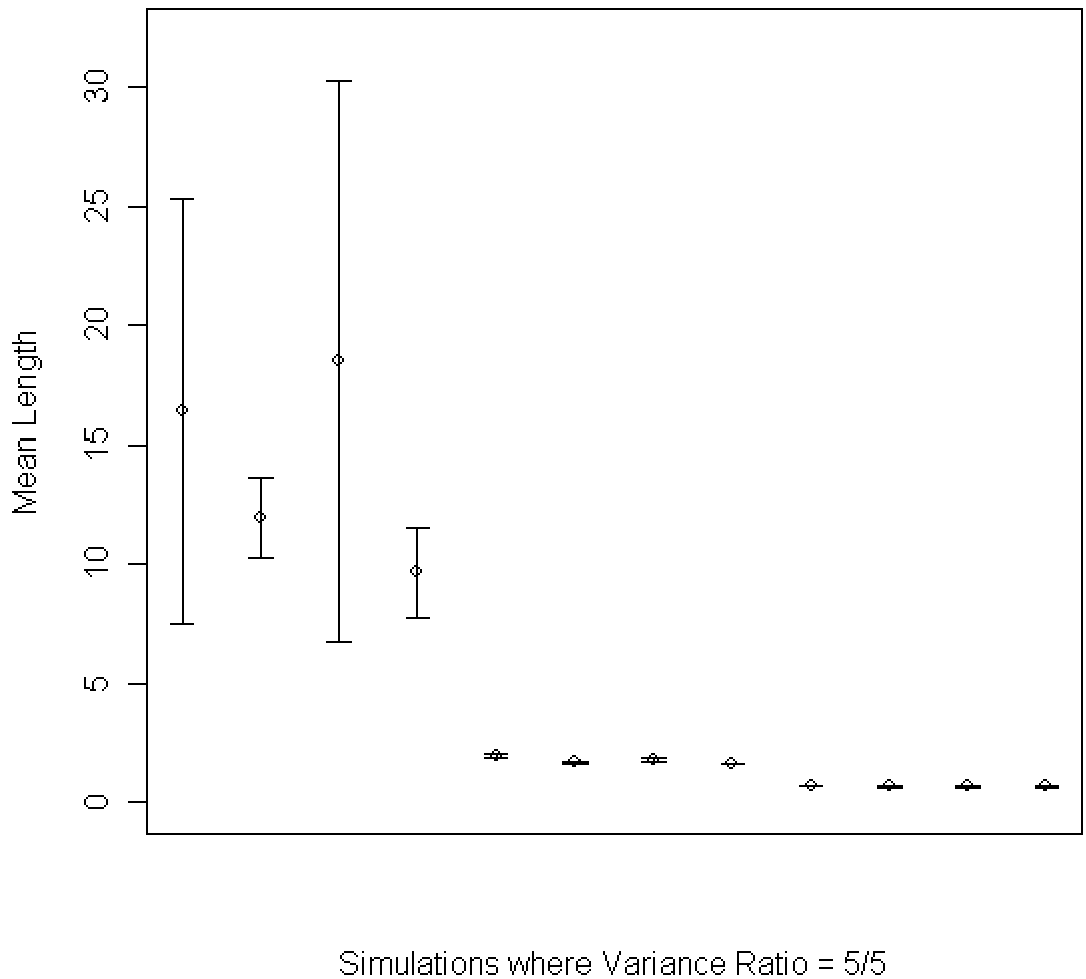


Figure D.11: Method 1: 95% confidence intervals for mean length when $\sigma_{\eta} / \sigma_{\varepsilon}$=5/.05

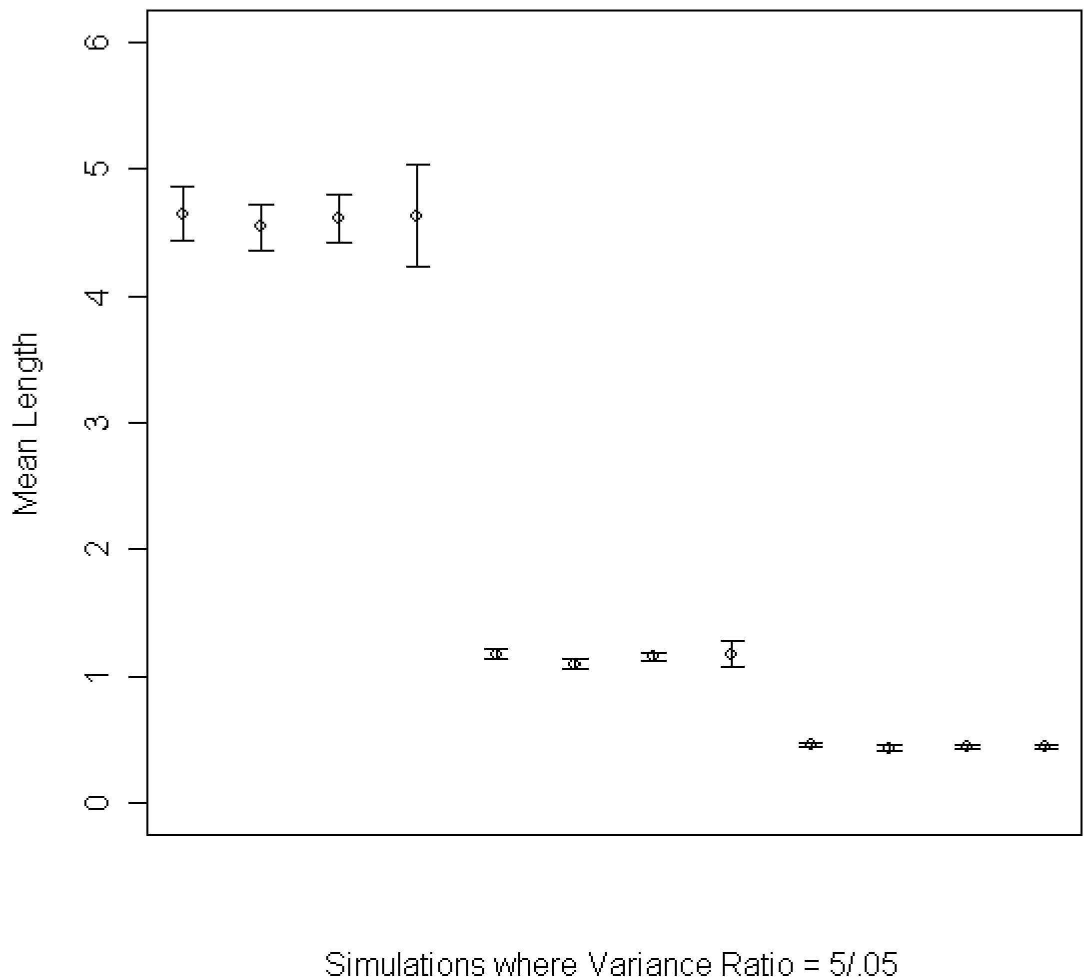
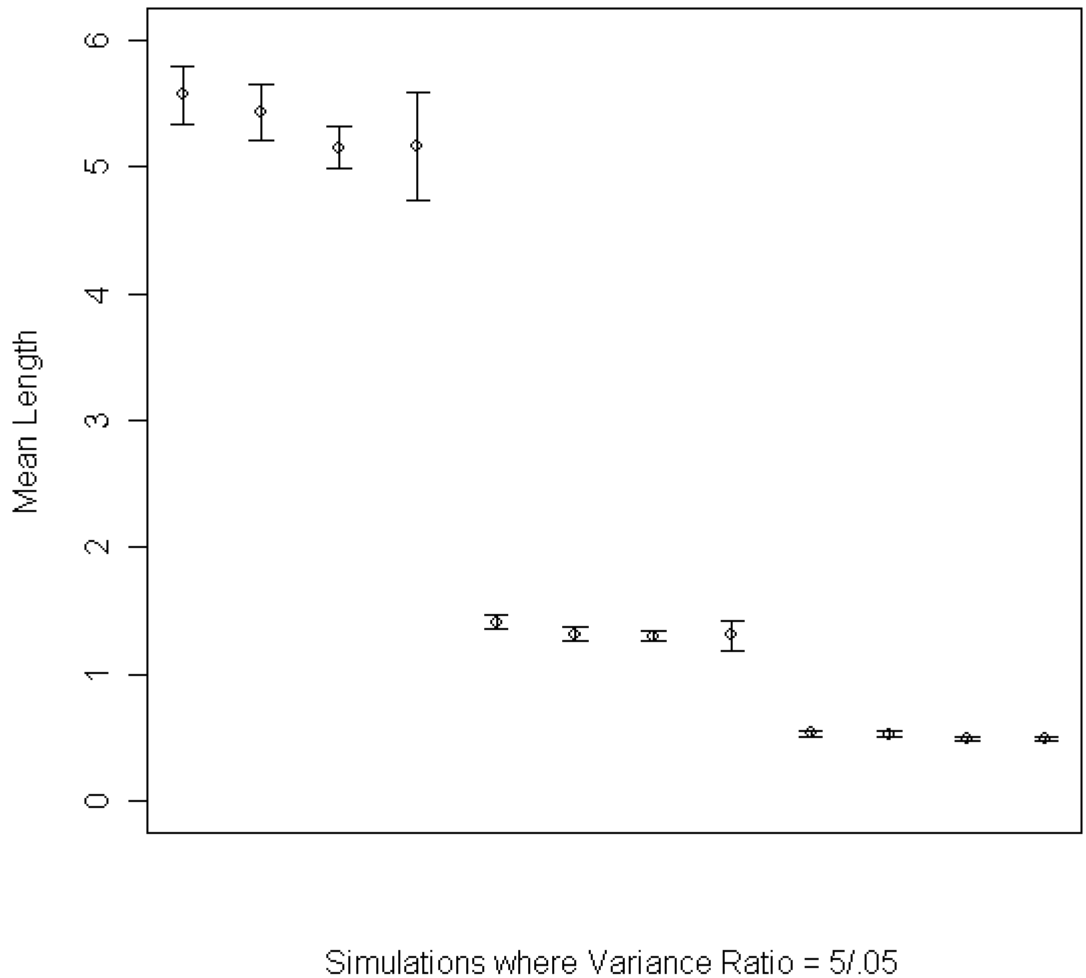


Figure D.12: Method 2: 95% confidence intervals for mean length when $\sigma_\eta / \sigma_\varepsilon$=5/.05



Appendix E- McNemar's Test

What follows are the 2x2 tables output by SAS's PROC FREQ and the result of McNemar's Test for each 2x2 table.

Count of Method 1 vs. Method 2 for sim: 11110

The FREQ Procedure

Frequency

Table of Method_1 by Method_2

Method_1	Method_2 Yes	Method_2 No	Total
Yes	75	6	81
No	0	3	3
Total	75	9	84

Frequency Missing = 2

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	6.0000
DF	1
Pr > S	0.0143

Simple Kappa Coefficient	
Kappa	0.4717
ASE	0.1765
95% Lower Conf Limit	0.1258
95% Upper Conf Limit	0.8176

Effective Sample Size = 84
Frequency Missing = 2

Count of Method 1 vs. Method 2 for sim: 11120

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	161	4	165
	No	1	9	10
	Total	162	13	175

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	1.8000
DF	1
Pr > S	0.1797

Simple Kappa Coefficient	
Kappa	0.7676
ASE	0.1000
95% Lower Conf Limit	0.5715
95% Upper Conf Limit	0.9637

Sample Size = 175

Count of Method 1 vs. Method 2 for sim: 11210

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	108	2	110
	No	1	5	6
	Total	109	7	116
	Frequency Missing = 5			

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.3333
DF	1
Pr > S	0.5637

Simple Kappa Coefficient	
Kappa	0.7556
ASE	0.1358
95% Lower Conf Limit	0.4895
95% Upper Conf Limit	1.0000

Effective Sample Size = 116
Frequency Missing = 5

Count of Method 1 vs. Method 2 for sim: 11220

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	181	4	185
	No	1	9	10
	Total	182	13	195
	Frequency Missing = 1			

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	1.8000
DF	1
Pr > S	0.1797

Simple Kappa Coefficient	
Kappa	0.7692
ASE	0.0995
95% Lower Conf Limit	0.5743
95% Upper Conf Limit	0.9642

Effective Sample Size = 195
Frequency Missing = 1

Count of Method 1 vs. Method 2 for sim: 12110

The FREQ Procedure

Frequency

Table of Method_1 by Method_2			
Method_1	Method_2		Total
	No	Yes	
No	9	1	10
Yes	3	105	108
Total	12	106	118
Frequency Missing = 6			

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	1.0000
DF	1
Pr > S	0.3173

Simple Kappa Coefficient	
Kappa	0.7997
ASE	0.0969
95% Lower Conf Limit	0.6097
95% Upper Conf Limit	0.9897

Effective Sample Size = 118
Frequency Missing = 6

Count of Method 1 vs. Method 2 for sim: 12120

The FREQ Procedure

Frequency	Table of Method_1 by Method_2		
Method_1	Method_2		Total
	Yes	No	
Yes	158	3	161
No	2	12	14
Total	160	15	175

Frequency Missing = 2

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.2000
DF	1
Pr > S	0.6547

Simple Kappa Coefficient	
Kappa	0.8120
ASE	0.0818
95% Lower Conf Limit	0.6517
95% Upper Conf Limit	0.9723

Effective Sample Size = 175
Frequency Missing = 2

Count of Method 1 vs. Method 2 for sim: 12210

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	91	3	94
	No	1	6	7
	Total	92	9	101
	Frequency Missing = 5			

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	1.0000
DF	1
Pr > S	0.3173

Simple Kappa Coefficient	
Kappa	0.7289
ASE	0.1289
95% Lower Conf Limit	0.4763
95% Upper Conf Limit	0.9815

Effective Sample Size = 101
Frequency Missing = 5

Count of Method 1 vs. Method 2 for sim: 12220

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	173	2	175
	No	3	13	16
	Total	176	15	191

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.2000
DF	1
Pr > S	0.6547

Simple Kappa Coefficient	
Kappa	0.8245
ASE	0.0766
95% Lower Conf Limit	0.6744
95% Upper Conf Limit	0.9746

Sample Size = 191

Count of Method 1 vs. Method 2 for sim: 13110

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		No	Yes	
	No	12	6	18
	Yes	0	182	182
	Total	12	188	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	6.0000
DF	1
Pr > S	0.0143

Simple Kappa Coefficient	
Kappa	0.7845
ASE	0.0846
95% Lower Conf Limit	0.6186
95% Upper Conf Limit	0.9503

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 13120

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		No	Yes	
	No	16	4	20
	Yes	0	180	180
	Total	16	184	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	4.0000
DF	1
Pr > S	0.0455

Simple Kappa Coefficient	
Kappa	0.8780
ASE	0.0599
95% Lower Conf Limit	0.7606
95% Upper Conf Limit	0.9955

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 13210

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	180	0	180
	No	4	16	20
	Total	184	16	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	4.0000
DF	1
Pr > S	0.0455

Simple Kappa Coefficient	
Kappa	0.8780
ASE	0.0599
95% Lower Conf Limit	0.7606
95% Upper Conf Limit	0.9955

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 13220

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	185	0	185
	No	3	12	15
	Total	188	12	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	3.0000
DF	1
Pr > S	0.0833

Simple Kappa Coefficient	
Kappa	0.8810
ASE	0.0677
95% Lower Conf Limit	0.7482
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 21110

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	191	2	193
	No	0	7	7
	Total	191	9	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	2.0000
DF	1
Pr > S	0.1573

Simple Kappa Coefficient	
Kappa	0.8699
ASE	0.0908
95% Lower Conf Limit	0.6920
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 21120

The FREQ Procedure

Frequency

Table of Method_1 by Method_2			
Method_1	Method_2		Total
	Yes	No	
Yes	191	1	192
No	1	7	8
Total	192	8	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.0000
DF	1
Pr > S	1.0000

Simple Kappa Coefficient	
Kappa	0.8698
ASE	0.0909
95% Lower Conf Limit	0.6915
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 21210

The FREQ Procedure

Frequency — Table of Method_1 by Method_2

Method_1	Method_2 Yes	Method_2 No	Total
Yes	189	2	191
No	2	7	9
Total	191	9	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.0000
DF	1
Pr > S	1.0000

Simple Kappa Coefficient	
Kappa	0.7673
ASE	0.1125
95% Lower Conf Limit	0.5468
95% Upper Conf Limit	0.9878

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 21220

The FREQ Procedure

Frequency

Table of Method_1 by Method_2			
Method_1	**Method_2**		**Total**
	Yes	**No**	
Yes	183	1	184
No	3	13	16
Total	186	14	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	1.0000
DF	1
Pr > S	0.3173

Simple Kappa Coefficient	
Kappa	0.8559
ASE	0.0707
95% Lower Conf Limit	0.7173
95% Upper Conf Limit	0.9945

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 22110

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	185	0	185
	No	3	12	15
	Total	188	12	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	3.0000
DF	1
Pr > S	0.0833

Simple Kappa Coefficient	
Kappa	0.8810
ASE	0.0677
95% Lower Conf Limit	0.7482
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 22120

The FREQ Procedure

Frequency

Table of Method_1 by Method_2			
Method_1	Method_2		Total
	Yes	No	
Yes	184	1	185
No	5	10	15
Total	189	11	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	2.6667
DF	1
Pr > S	0.1025

Simple Kappa Coefficient	
Kappa	0.7536
ASE	0.0964
95% Lower Conf Limit	0.5647
95% Upper Conf Limit	0.9425

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 22210

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	193	1	194
	No	1	5	6
	Total	194	6	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.0000
DF	1
Pr > S	1.0000

Simple Kappa Coefficient	
Kappa	0.8282
ASE	0.1193
95% Lower Conf Limit	0.5944
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 22220

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	189	1	190
	No	1	9	10
	Total	190	10	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.0000
DF	1
Pr > S	1.0000

Simple Kappa Coefficient	
Kappa	0.8947
ASE	0.0737
95% Lower Conf Limit	0.7502
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 23110

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	182	0	182
	No	10	8	18
	Total	192	8	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	10.0000
DF	1
Pr > S	0.0016

Simple Kappa Coefficient	
Kappa	0.5928
ASE	0.1146
95% Lower Conf Limit	0.3682
95% Upper Conf Limit	0.8175

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 23120

The FREQ Procedure

Table of Method_1 by Method_2 (Frequency)

Method_1	Method_2 Yes	Method_2 No	Total
Yes	182	0	182
No	7	11	18
Total	189	11	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	7.0000
DF	1
Pr > S	0.0082

Simple Kappa Coefficient	
Kappa	0.7409
ASE	0.0929
95% Lower Conf Limit	0.5588
95% Upper Conf Limit	0.9230

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 23210

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	186	0	186
	No	2	11	13
	Total	188	11	199

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	2.0000
DF	1
Pr > S	0.1573

Simple Kappa Coefficient	
Kappa	0.9114
ASE	0.0621
95% Lower Conf Limit	0.7896
95% Upper Conf Limit	1.0000

Sample Size = 199

Count of Method 1 vs. Method 2 for sim: 23220

The FREQ Procedure

Frequency	Table of Method_1 by Method_2		
Method_1	Method_2		Total
	Yes	No	
Yes	184	0	184
No	1	15	16
Total	185	15	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	1.0000
DF	1
Pr > S	0.3173

Simple Kappa Coefficient	
Kappa	0.9650
ASE	0.0349
95% Lower Conf Limit	0.8967
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 31110

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	186	2	188
	No	1	11	12
	Total	187	13	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.3333
DF	1
Pr > S	0.5637

Simple Kappa Coefficient	
Kappa	0.8720
ASE	0.0729
95% Lower Conf Limit	0.7292
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 31120

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	188	0	188
	No	0	12	12
	Total	188	12	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	.
DF	1
Pr > S	.
NOTE: There are no discordant pairs.	

Simple Kappa Coefficient	
Kappa	1.0000
ASE	0.0000
95% Lower Conf Limit	1.0000
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 31210

The FREQ Procedure

Frequency — Table of Method_1 by Method_2

Method_1	Method_2 Yes	Method_2 No	Total
Yes	187	2	189
No	2	9	11
Total	189	11	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.0000
DF	1
Pr > S	1.0000

Simple Kappa Coefficient	
Kappa	0.8076
ASE	0.0938
95% Lower Conf Limit	0.6238
95% Upper Conf Limit	0.9914

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 31220

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	192	0	192
	No	2	6	8
	Total	194	6	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	2.0000
DF	1
Pr > S	0.1573

Simple Kappa Coefficient	
Kappa	0.8521
ASE	0.1029
95% Lower Conf Limit	0.6503
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 32110

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	186	1	187
	No	1	12	13
	Total	187	13	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.0000
DF	1
Pr > S	1.0000

Simple Kappa Coefficient	
Kappa	0.9177
ASE	0.0577
95% Lower Conf Limit	0.8046
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 32120

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	192	0	192
	No	2	6	8
	Total	194	6	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	2.0000
DF	1
Pr > S	0.1573

Simple Kappa Coefficient	
Kappa	0.8521
ASE	0.1029
95% Lower Conf Limit	0.6503
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 32210

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	185	1	186
	No	3	11	14
	Total	188	12	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	1.0000
DF	1
Pr > S	0.3173

Simple Kappa Coefficient	
Kappa	0.8355
ASE	0.0805
95% Lower Conf Limit	0.6778
95% Upper Conf Limit	0.9933

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 32220

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	184	1	185
	No	2	13	15
	Total	186	14	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	0.3333
DF	1
Pr > S	0.5637

Simple Kappa Coefficient	
Kappa	0.8885
ASE	0.0636
95% Lower Conf Limit	0.7638
95% Upper Conf Limit	1.0000

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 33110

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	No	8	10	18
	Yes	182	0	182
	Total	190	10	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	154.0833
DF	1
Pr > S	<.0001

Simple Kappa Coefficient	
Kappa	-0.1047
ASE	0.0337
95% Lower Conf Limit	-0.1708
95% Upper Conf Limit	-0.0387

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 33120

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	180	0	180
	No	11	9	20
	Total	191	9	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	11.0000
DF	1
Pr > S	0.0009

Simple Kappa Coefficient	
Kappa	0.5956
ASE	0.1084
95% Lower Conf Limit	0.3831
95% Upper Conf Limit	0.8081

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 33210

The FREQ Procedure

Frequency — **Table of Method_1 by Method_2**

Method_1	Method_2 Yes	Method_2 No	Total
Yes	185	0	185
No	5	10	15
Total	190	10	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	5.0000
DF	1
Pr > S	0.0253

Simple Kappa Coefficient	
Kappa	0.7872
ASE	0.0918
95% Lower Conf Limit	0.6073
95% Upper Conf Limit	0.9672

Sample Size = 200

Count of Method 1 vs. Method 2 for sim: 33220

The FREQ Procedure

Frequency	Table of Method_1 by Method_2			
	Method_1	Method_2		Total
		Yes	No	
	Yes	182	0	182
	No	9	9	18
	Total	191	9	200

Statistics for Table of Method_1 by Method_2

McNemar's Test	
Statistic (S)	9.0000
DF	1
Pr > S	0.0027

Simple Kappa Coefficient	
Kappa	0.6454
ASE	0.1080
95% Lower Conf Limit	0.4337
95% Upper Conf Limit	0.8571

Sample Size = 200