KERNEL-BASED INTERVAL ESTIMATES OF INVERSE DOSE RESPONSE WITH APPLICATIONS TO GENETIC CLINES

By

Annie Y. Lo

Submitted to the

Faculty of the College of Arts and Sciences

of American University

in Partial Fulfillment of

the Requirements for the Degree of

Doctor of Philosophy

In

Statistics

Chair:

Robert W. Jernigan, Ph.D.

John P. Nolan, Ph.D.

Monica C. Jackson, Ph.D.

Dean of the College of Arts and Sciences

April 21, 2011

Date

2011
American University
Washington, D.C. 20016

KERNEL-BASED INTERVAL ESTIMATES OF INVERSE DOSE RESPONSE WITH APPLICATIONS TO GENETIC CLINES

BY

Annie Y. Lo

ABSTRACT

We investigate nonparametric regression techniques to estimate the distribution of the $LD_{100\alpha}$, $0 < \alpha < 1$, the lethal dose where 100α % of subjects show a response. Kernel methods are used to estimate the resulting response probability curve using real and simulated data. We apply and extend these kernel-based estimation procedures to a problem in evolutionary genetics where the prevalence of a genetic trait is mapped. In this setting, distance serves as a dose and the response probability curve is called a cline. We investigate the distributional properties of kernel estimates of $LD_{100\alpha}$ with special attention to the LD_{20}, LD_{80}, and the distance between them, called the cline width. Confidence intervals are constructed for LD_{20}, LD_{80}, and the cline width and small sample properties are investigated through series expansion and simulation.

KERNEL-BASED INTERVAL ESTIMATES OF INVERSE DOSE RESPONSE WITH APPLICATIONS TO GENETIC CLINES

BY

Annie Y. Lo

ABSTRACT

We investigate nonparametric regression techniques to estimate the distribution of the $LD_{100\alpha}$, $0 < \alpha < 1$, the lethal dose where 100α % of subjects show a response. Kernel methods are used to estimate the resulting response probability curve using real and simulated data. We apply and extend these kernel-based estimation procedures to a problem in evolutionary genetics where the prevalence of a genetic trait is mapped. In this setting, distance serves as a dose and the response probability curve is called a cline. We investigate the distributional properties of kernel estimates of $LD_{100\alpha}$ with special attention to the LD_{20}, LD_{80}, and the distance between them, called the cline width. Confidence intervals are constructed for LD_{20}, LD_{80}, and the cline width and small sample properties are investigated through series expansion and simulation.

ACKNOWLEDGEMENTS

My deepest gratitude and thanks to:

My advisor, Dr. Robert Jernigan for his excellent guidance, inspiration, insight, and compassion. He has helped him grow, learn, and discover. His “No Whining” button on his bulletin board outside his office constantly reminds me to not make excuses, but to be a responsible and diligent person. He is my mentor and role model.

My committee members, Dr. John Nolan and Dr. Monica Jackson for their thorough review of the dissertation, helpful comments, time, and attention. Special thanks to Dr. Nolan for his lovely poem which he presented on the day of the dissertation defense:

> Annie has written a thesis on clines,
> which are not the same things as splines.
> This has caused her family to whine,
> and her to consume glasses of wine.
> But soon all will be fine,
> when the committee considers and signs.

Our Department Chair, Dr. Mary Gray and the Mathematics and Statistics faculty and staff for providing a first class research environment; Dr. I-Lok Chang for being such an outstanding professor; and Linda Greene for all her administrative assistance.

My colleagues at Westat for their encouragement and support. Special thanks go to David Morganstein who values professional development for his staff members and

renders me the opportunity to fulfill my dreams; to Margo Tercy who has significantly contributed to formatting the manuscript.

My family and friends for their love and care, especially to my husband and daughters who give me hope to carry on; to my parents who believe in me, to my brothers who stand by me, and to my aunt, Virginia Mok who opened the door of higher education to me. This dissertation is dedicated to my husband, Bernie who has created a sweet and comfortable home for me to concentrate on my research, and to my daughters, Andrea, Sarah, and Kristi for their patience, understanding, and all their good luck wishes.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF ILLUSTRATIONS

Figure

CHAPTER 1

INTRODUCTION

The motivation of this dissertation stems from a research problem in evolutionary genetics in describing the geographic patterns of genetic diversity in a species. For example, Jaarola et al. (1997) mapped the geographic prevalence of certain genetic markers of voles. The prevalence of the genetic marker in a population is modeled by a monotonic sigmoidal-shaped curve $p(x)$ dependent on a distance x from a fixed point. This represents the probability of a response and in the biological literature, such a curve is called a cline. It is in essence a dose response curve with dose corresponding here to distance. Biological assays are commonly used to study the dose response of the toxic effects of a chemical. The analysis often results in a similar sigmoidal-shaped dose-response curve $p(x)$ for a dose level x, showing that the toxic effects increase with increasing levels of the doses. The observed reaction y_i of the ith subject ($i = 1, \ldots, n$) at dose level x_i is assumed to follow a Bernoulli $(1, p(x_i))$ distribution, so that $y_i = 1$ represents the presence of a response and $y_i = 0$ represents the absence of the response. Of interest is the estimate of $p(x)$ and functionals of $p(x)$, $LD_{100\alpha}$, for $0 < \alpha < 1$, the lethal dosage (LD) level at which 100α% of the population dies. Of primary interest to biomedical researchers is the estimation of the LD_{50}. In the area of carcinogenic risk assessment, values of α less than 0.5 are also of interest.

The structure of a cline as a dose response curve in evolutionary biology yields information about gene flow and speciation mechanisms. Evolutionary biologists are interested in the extent of similarity of clines estimated by different genetic markers (Brumfield et al., 2001). Of concern are measures of the location and width of the cline. The location of the cline is defined as the LD_{50}, here the distance that corresponds to a 50% prevalence of the marker studied. The cline width is the distance between the LD_{20} and the LD_{80}, here the distances that correspond to a 20% and 80% marker prevalence. These measures provide valuable information to biologists regarding the extent of a zone where two species might hybridize, yielding crucial data for genetic differentiation.

Dose-response curves can be modeled parametrically and nonparametrically. Common parametric approaches, shown in detail below, include the probit models (Bliss, 1934; Finney, 1978) and logistic regression (Berkson, 1944). Various nonparametric methods for estimating the $LD_{100\,\alpha}$ of the dose-response curve are also briefly reviewed below. These include the Spearman-Karber estimator and the trimmed Spearman-Karber estimator (Hamilton, 1979; James, James, and Westenberger, 1984), robust estimators such as the *L*, *M*, and *R*-estimators (Miller and Halpern, 1980; James, James, and Westenberger, 1984) and smoothing splines (Brumfield et al., 2001). We will conclude the review with kernel estimators and methods (Müller and Schmitt, 1988; Hart, 1997) that we will examine in much more detail in the following chapters.

The probit model assumes that $p(x)$ is a normal cumulative distribution function. The probit curve p_1 is $p_1(x) = \Phi\left(\frac{x-\mu}{\sigma}\right)$, where Φ is the normal cumulative distribution

function, μ is the location, and σ is the slope parameter. The probit has the following properties:

$$\log LD_{50} = \mu \text{ and}$$

$$p_1'(\log LD_{50}) = \frac{1}{\sqrt{2\pi}\sigma}.$$

Maximum likelihood estimators $\hat{\mu}$ and $\hat{\sigma}$ can be computed from the following likelihood function:

$$L(\mu, \sigma) = \prod_{i=1}^{n} p_1(x_i)^{y_i} (1 - p_1(x_i))^{1-y_i} .$$

Dose response curves have been traditionally modeled also using logistic regression. In this approach the frequency $p(x)$ at distance x, is modeled as:

$$p(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}},$$

where α and β are parameters to be estimated. The maximum likelihood estimates $\hat{\alpha}$ and $\hat{\beta}$ are obtained by maximizing the log-likelihood

$$l(\alpha, \beta) = \sum_{i=1}^{n} \{y_i \log[p(x_i)] + (1 - y_i) \log[1 - p(x_i)]\}. \qquad (1.1)$$

To fit the logistic regression model, the frequencies are transformed to the form:

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \alpha + \beta x .$$

The transformed frequencies $\log\left(\frac{p(x)}{1-p(x)}\right)$ are called the logits of $p(x)$. These logit values can be fit by maximum likelihood using an iterative weighted linear regression on distance d (McCullagh and Nelder, 1989).

Nonparametric methods designed for estimating the LD_{50} were first proposed by Spearmen (1908) and Thompson (1947). Traditional nonparametric techniques for estimating only LD_{50} include the Spearman-Karber estimator and the trimmed Spearman-Karber estimator (Hamilton, 1979; James, James, and Westenberger, 1984). When the mean of the population tolerance distribution exists, the Spearman-Karber estimator is the mean of the empirical tolerance distribution defined by the following:

$$\hat{\theta}_{SK} = \int_{-\infty}^{\infty} x \, d\tilde{F}(x),$$

where $\tilde{F}(x)$ is the empirical response curve. If the observed proportion of responses to first dose x_1 is $p_1 = 0$ and the response proportion of the last dose x_k is $p_k = 1$, then the Spearman-Kaber estimator can be written as

$$\hat{\theta}_{SK} = \sum_{i=1}^{k} (p_{i+1} - p_i)(x_{i+1} + x_i)/2.$$

The 100α%-trimmed Spearman-Karber estimator is defined by trimming 100α% from each tail of the empirical tolerance distribution and taking the mean of the (appropriately normalized) remaining central part of the distribution:

$$\hat{\theta}_{SK100\alpha\%} = \frac{\int_{\tilde{F}^{-1}\alpha}^{\tilde{F}^{-1}(1-\alpha)} x \, d\tilde{F}(x)}{1-2\alpha},$$

$0 \leq \alpha < 0.5$. The 50%-trimmed Spearman-Karber estimator is the median, which is the limit of the 100α%-trimmed Spearman-Karber estimator as $\alpha \rightarrow 0.5$. These nonparametric Speaman estimators require the strong assumption of point symmetry of $p(x)$. That is, $p(\log LD_{50} - x) = 1 - p(\log LD_{50} + x)$.

Other nonparametric methods for estimating LD_{50} include robust estimators. Different classes of robust estimators are the *L*, *M*, and *R*-estimators. *L*-estimators are linear combinations of order statistics. Miller and Halpern (1980) defined *L*- and *M*-estimators of the LD_{50} for quantal bioassay to be explicit or implicit functionals of the empirical tolerance distribution, the functionals being the same as those applied to the empirical distribution function in the *iid* case. The *L*-estimator for quantal bioassay is defined by:

$$\hat{\theta}_L = x_0 + \Delta x \sum_{i=-k}^{k+1} iJ(\hat{p}_i)(\hat{p}_i - \hat{p}_{i-1}),$$

where Δx is the equal dose spacing, x_0 is the middose, $\hat{p}_{-k-1} \equiv p_{-k-1} = 0$ and $\hat{p}_{k+1} \equiv p_{k+1} = 1$ by definition. The function $J(u)$ is defined on the interval [0,1] and is symmetric about ½ with $\int_0^1 J(u)du = 1$.

M-estimators are maximum likelihood type estimators. The *M*-estimator, $\hat{\theta}_M$ for quantal bioassay (Miller and Halpern, 1980) is defined to be the root of the equation:

$$\sum_{i=-k}^{k+1} \Psi(x_i - \Theta)(\hat{p}_i - \hat{p}_{i-1}) = 0, \qquad (1.2)$$

where $\hat{p}_{-k-1} \equiv p_{-k-1} = 0$ and $\hat{p}_{k+1} \equiv p_{k+1} = 1$ by definition. The Ψ-function is a generalization of the function $-f'(x)/f(x)$, which gives the maximum likelihood estimator for a location parameter θ. Iterative algorithms are necessary to find the root of (1.2). If the Ψ-function is non-monotone, problems can arise with the uniqueness of the root of (1.2) and the convergence of the iterative procedure.

R-estimators are estimators derived from rank tests. James, James, and Westenberger (1984) defined *R*-estimators for the LD_{50} in quantal bioassay. Let *J* be a nondecreasing integrable function defined on (0, 1), such that $J(1\text{-}t) = -J(t)$ and *J* is not identically equal to zero. The *R*-estimator $\hat{\theta}$ based on *J* is the solution of the equation

$$\int_{-\infty}^{\infty} J\left(\frac{\tilde{F}(x)+1-\tilde{F}(2\theta-x)}{2}\right) d\tilde{F}(x) = 0.$$

Müller and Schmitt (1988) compared the probit maximum likelihood estimator and kernel estimators in quantal bioassay for effective doses $LD_{100\alpha}$ where α = .01, .05, .10, and .50. For the kernel method, they presented two asymptotically consistent approaches for constructing confidence intervals for the estimated $LD_{100\alpha}$. For constructing confidence intervals for LD_{50}, Kelly (2001) presented three methods – Fieller's method, profile likelihood, and the bootstrap. In the simulation, none of the three methods were found to be completely satisfactory for providing confidence intervals for LD_{50} unless very large sample sizes are taken.

The most recent work on genetic cline estimation is that of Brumfield et al. (2001) using smoothing splines. Smoothing splines are flexible curves with various degrees of smoothness specified by a smoothing parameter. These curves are developed with an overall view of minimizing a penalized likelihood function. In this approach a log-likelihood similar to that given by Equation (1.1) in the logistic regression discussion is penalized with a measure of roughness of the resulting fitted curve. The function *p* is chosen that minimizes the negative penalized log-likelihood function

$$-\sum_{i=1}^{n} l(p(d_i)) + n\lambda \int \left[p''(d)\right]^2 dd$$

Here the standard likelihood is penalized for lack of smoothness. The integral measures the "roughness" of the chosen function *p*. A rough function has a rapidly changing slope. The overall rate of change of the slope of *p* can be measured by the integration of the square of its second derivative. The parameter λ is called the smoothing parameter which can also be indexed by an equivalent degrees of freedom. These equivalent degrees of freedom indicate the number of parameters needed to specify a function of the desired roughness or smoothness. Specifying values for either governs how much smoothness or roughness is permitted in the resulting function *p*. As $\lambda \rightarrow \infty$ larger penalties for roughness are imposed, the degrees of freedom approach 2 indicating that two parameters (a slope and an intercept) are needed. The resulting *p* that minimizes the negative penalized log-likelihood becomes smoother, with its logit approaching a straight line. As $\lambda \rightarrow 0$, and little penalty is imposed for roughness, the degrees of freedom grow, the minimizing *p* can be rougher, eventually interpolating the data points. Values of λ between these extremes produce smooth curves that can closely model the patterns in the data.

In the approach of Brumfield et al. (2001) the presence of a strong monotonic pattern of many genetic markers and the belief that a smooth monotonic function best describes the cline, often results in an assumption of monotonicity of the fitted cline. Algorithmically, as the equivalent degrees of freedom are reduced from a large value down to the value of 2, the resulting smoothing spline fits go from being a non-monotonic interpolating function to a monotonic logistic fit (specified by a slope and intercept). The approach of Brumfield et al. (2001) was to examine the largest degrees of freedom that results in a monotonic fitting function.

Smoothing splines estimate the underlying genetic cline in a way that allows for an explicit consideration of the overall goodness of fit of the cline to the data. As flexible as this approach can be, its solution must come from a computational minimization procedure. An explicit formula is generally not available. Kernel estimation methods provide for this explicit estimation approach. These methods estimate the cline at a given point by a weighted average of "local" observations (Green and Silverman, 1994). We will introduce these kernel methods as developed by Müller and Schmitt (1988) and Hart (1997). Estimates of the response function and asymptotic distributional properties for the functionals LD_{20}, LD_{80}, and the cline width will be presented.

Chapter 2 of this dissertation provides the theoretical background of kernel estimation. Based on the distributional properties of the kernel estimator, we developed the distributional properties of LD_{20}, LD_{80}, and the cline width. Chapter 3 presents seven methods for computing the confidence intervals for the kernel estimates. The first two methods are adapted from Müller and Schmitt (1988). Müller and Schmitt assumed that the bias could be neglected when computing confidence intervals. In the third method, we extended the results of Müller and Schmitt by correcting the confidence intervals for bias. The next four methods are based on the distributional properties developed in Chapter 2.

To examine the distributional results and to evaluate the performance of the kernel estimator for LD_{20}, LD_{80}, and the cline width, we performed a simulation study. Through generating replicate samples from the probit data, we examined the distributional properties of the kernel estimates. For various sample sizes, distributions of distance, and kernels, we applied each of the seven methods presented in Chapter 3 to compute the corresponding 95% confidence intervals for the estimates. The performance

review was based on the average length of the confidence intervals, the midpoint of the intervals, and the coverage probabilities. Results of the simulations are shown in Chapter 4.

We applied the kernel approach to estimate LD_{20}, LD_{80}, and the cline width for the field vole genetic data of Jaarola (1997). The results of the estimates and their corresponding 95% confidence intervals are presented in Chapter 5. We conclude our findings in the last chapter, where we discuss topics for future research.

CHAPTER 2

KERNEL ESTIMATION

The underlying assumption for our model is that the observed trait y_i of the ith individual ($i = 1,\ldots, n$) at distance x_i follows a Bernoulli distribution with probability $p(x_i)$ given by

$$\begin{cases} P(y_i = 1) = p(x_i) \\ P(y_i = 0) = 1 - p(x_i). \end{cases}$$

Distances x_i's are assumed to have density $f(x)$. The Bernoulli trails are assumed to be independent for different individuals. The function p is the dose-response curve. Our goal is to estimate the distribution of LD_{20} and LD_{80} where 20% and 80% of the individuals possess a given trait. We are also interested in estimating the distribution of the distance between LD_{20} and LD_{80} called the cline width. We use a nonparametric kernel method in the estimation.

Kernel estimation is a nonparametric smoothing technique. Kernel estimators smooth out the contribution of each observed data point over a local neighborhood of that data point. We consider the Gasser and Müller kernel estimator in our study, defined as

$$\hat{p}(x) = \frac{1}{h}\sum_{i=1}^{n} y_i \int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right) du \qquad (2.1)$$

where h is called the bandwidth that controls the smoothness of $\hat{p}(x)$; K is a function called the kernel; $\frac{1}{h}\int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right) du$ is the weight function; and $s_0 = 0$, $s_i = \frac{x_i + x_{i+1}}{2}$,

$s_n = 1$. The distances x_i are assumed to be uniformly distributed, that is,

$x_i = (i - 1) / (n - 1)$, $i = 1, \ldots, n$.

The following conditions of the kernel K are required:

- K has support (-1, 1);
- K is continuous;
- $\int_{-1}^{1} K(u)\, du = 1$;
- $\int_{-1}^{1} uK(u)\, du = 0$.

These conditions are necessary to have desirable consistency properties for the mean, variance, and bias. Epanechnikov (1969) found an optimal kernel by minimizing the mean squared error of the Gasser and Müller kernel estimator subject to the constraints that K has finite support and zero first moment. This optimal kernel $K(u) = \frac{3}{4}\left(1 - u^2\right)$ for $-1 \le u \le 1$ and 0 otherwise is known as the Epanechnikov kernel.

Distributional Properties of Kernel Estimator

To simplify the presentation of formulas, we define

$$J_K = \int_{-1}^{1} K^2(u)\, du \quad \text{and} \quad \sigma_K^2 = \int_{-1}^{1} u^2 K(u)\, du .$$

The expected value of the Gasser and Müller kernel estimator (Gasser and Müller 1979; Hart 1997) can be represented as

$$E(\hat{p}(x)) = \frac{1}{h}\int_0^1 p(u) K\left(\frac{x-u}{h}\right) du + O\left(\frac{1}{n}\right),$$

where

$$O\left(\frac{1}{n}\right) = \frac{1}{h}\sum_{i=1}^{n}\int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right)[p(x_i) - p(u)]\,du$$

are terms of order 1/*n*. The variance of the kernel estimator (Gasser and Müller 1979; Hart 1997) can be represented as

$$Var(\hat{p}(x)) = \frac{\sigma^2}{nh}\frac{1}{f(x)}J_K + O\left(\frac{1}{n}\right) + O\left(\frac{1}{n^2h^2}\right) \qquad (2.2)$$

In $Var(\hat{p}(x))$,

$$O\left(\frac{1}{n}\right) = \frac{\sigma^2}{nh}\left[\int_{-1}^{1}\frac{1}{f(x-hz)}K^2(z)dz - \frac{1}{f(x)}\int_{-1}^{1}K^2(z)dz\right] \text{ and}$$

$$O\left(\frac{1}{n^2h^2}\right) = \frac{\sigma^2}{nh^2}\sum_{i=1}^{n}(s_i - s_{i-1})\left[\frac{1}{f(x_i^*)}K^2\left(\frac{x-x_i^*}{h}\right) - \frac{1}{f(x'_i)}K^2\left(\frac{x-x'_i}{h}\right)\right],$$

x_i^* and $x'_i \in [s_{i-1}, s_i], i = 1,\text{K}\ ,n.$

From this representation we can see that for the variance to tend to 0, we need *n* to tend to ∞. Also, a smaller *h* means that we have fewer design points to be averaged, and therefore results in a larger variance. Equation (2.2) indicates that even if *h* is small, the variance will tend to 0 when *nh* tends to ∞. The design density *f*(*x*) also plays a role in the size of the variance; the variance will be small when the density of the design points is large. The bias of the Gasser-Müller estimator (Gasser and Müller 1979; Hart 1997) can be represented as

$$E(\hat{p}_h(x)) - p(x) = \frac{h^2}{2}p''(x)\sigma_K^2 + o(h^2) + O(n^{-1}) \qquad (2.3)$$

with

$$O(n^{-1}) = \frac{1}{h}\sum_{i=1}^{n}\int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right)[p(x_i) - p(u)]\,du\,.$$

For the bias to be zero, it is necessary for h to tend to 0. For σ_K^2 positive, the kernel estimator tends to underestimate $p(x)$ when $p''(x)$ is negative (peaked at x). Conversely, the kernel estimators overestimates $p(x)$ when $p''(x)$ is positive. The bias is largest at the sharpest peak or valley. The fact that the bias does not depend on the design density, $f(x)$ makes the Gasser-Müller kernel estimator more appealing than other types of Kernel estimators such as the Nadaraya-Watson estimator (Nadaraya, 1964 and Watson, 1964). From Equations (2.2) and (2.3), we have

$$Var((nh)^{1/2}\,\hat{p}(x)) = \frac{\sigma^2}{f(x)} J_K + O\left(\frac{1}{n}\right) + O\left(\frac{1}{n^2h^2}\right) \text{ and}$$

$$E((nh)^{1/2}(\hat{p}_h(x)) - p(x)) = \frac{n^{1/2}h^{5/2}}{2} p''(x)\sigma_K^2 + o(h^2) + O(n^{-1}).$$

As stated in Müller and Schmitt (1988), if $nh^5 \to \tau^2$ as $n \to \infty$ for some $\tau \geq 0$ then

$$(nh)^{\frac{1}{2}}(\hat{p}(x) - p(x)) \xrightarrow{D} N\left(\tau p''(x)\frac{\sigma_K^2}{2},\ \frac{p(x)(1-p(x))J_K}{f(x)}\right). \qquad (2.4)$$

Distributional Properties of $LD_{100\,\alpha}$

Based on the known distributional properties of the Kernel estimator, we follow Müller and Schmitt (1988) and Hart (1997) and derive the distributional properties of $LD_{100\,\alpha}$. From the distribution of $\hat{p}(x)$ in (2.4), we define $Z_{n,h}$ which from standard results asymptotically follows a standard normal distribution:

$$Z_{n,h} = \frac{\hat{p}_h(x) - E[\hat{p}_h(x)]}{\sqrt{Var[\hat{p}_h(x)]}} \xrightarrow{D} N(0,1).$$

Define

$$B_{n,h} = \frac{E[\hat{p}_h(x)] - p(x)}{\sqrt{Var[\hat{p}_h(x)]}},$$

and we have the following the quantity

$$\frac{\hat{p}_h(x) - p(x)}{\sqrt{Var[\hat{p}_h(x)]}} = Z_{n,h} + B_{n,h}.$$

The numerator in $B_{n,h}$ is the bias of the kernel estimator as given in Equation (2.3). The square of the denominator in $B_{n,h}$ is computed as

$$Var(\hat{p}_h(x)) = \sigma^2 \sum_{i=1}^{n} \left(\frac{1}{h} \int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h} \right) du \right)^2. \qquad (2.5)$$

By minimizing the mean squared error of the kernel estimator (Hart, Corollary 3.1), we obtain the optimal local bandwidth choice, *h** as follows. The mean squared error of the kernel estimator (Hart) is

$$M(x;h) = \tilde{M}(x;h) + R(n,h),$$

where

$$\tilde{M}(x;h) = \frac{\sigma^2}{nh} \frac{1}{f(x)} J_k + \frac{h^4}{4} (p''(x))^2 \sigma_k^4$$

and

$$R(n,h) = o(h^4) + O(n^{-1}) + O(nh)^{-2}.$$

The first term in $\tilde{M}(x;h)$ is the variance as shown in Equation (2.2); the second term is the squared bias as shown in Equation (2.3). When *h* decreases, the variance increases but the squared bias decreases. To balance the contribution of the variance and the squared bias to the mean squared error, *h* can be chosen to minimize *M*(*x*;*h*). To find the optimal

local bandwidth, we minimize the dominant term $\tilde{M}(x;h)$ by differentiating $\tilde{M}(x;h)$ with respect to *h*, setting the derivative to zero, and solving for *h*. We obtain

$$\frac{d}{dh}\tilde{M}(x;h) = -\frac{\sigma^2}{nf(x)}J_k + h^5(p''(x))^2\sigma_k^4 = 0.$$

Solving for *h* leads to the following optimal local bandwidth:

$$h^* = \left(\frac{\sigma^2 J_K}{f(x)[p''(x)]^2\sigma_K^4}\right)^{1/5} n^{-1/5}. \tag{2.6}$$

Asymptotically, $\sigma^2 \to \alpha(1-\alpha)$ as $n \to \infty$. The optimal bandwidth is proportional to $\sigma^{\frac{2}{5}}$, the more variation in the data, the larger the optimal bandwidth. The optimal bandwidth is inversely proportional to the fifth root of the design density, the curvature, and the sample size. When the design is dense, the optimal bandwidth is small. When *p*(*x*) has a lot of curvature, the second derivative of *p*(*x*) changes very rapidly; we need to average the data in a smaller neighborhood, resulting in a small optimal bandwidth. When the sample size is large, the optimal bandwidth is relatively small.

Let $C = \left(\frac{\alpha(1-\alpha)J_K}{f(x)[p''(x)]^2\sigma_K^4}\right)^{1/5}$. Taking $h = Cn^{-1/5}$ and using Equations (2.3) and (2.5), $B_{n,h}$ becomes

$$B_{n,h} = \frac{C^3 n^{-3/5}\sigma_K^2 p''(x) + o(C^2 n^{-2/5}) + O(n^{-1})}{2\sigma\left(\sum_{i=1}^{b}\left\{\int_{s_{i-1}}^{s_i} K\left[n^{1/5}(x-u)/C\right]du\right\}^2\right)^{1/2}}. \tag{2.7}$$

We will estimate the quantity $B_{n,h}$ for constructing the confidence interval.

Ignoring the terms of $O\left(\frac{1}{n}\right)$ and smaller, $B_{n,h}$ is asymptotic to

$$B_{C,n} = \frac{C^3 n^{-3/5} \sigma_K^2 p''(x)}{2\sigma\left(\sum_{i=1}^{b} \left\{ \int_{s_{i-1}}^{s_i} K\left[n^{1/5}(x-u)/C \right] du \right\}^2 \right)^{1/2}}$$

and

$$\frac{\hat{p}_h(x) - p(x)}{\sqrt{Var[\hat{p}(x)]}} - B_{C,n} \xrightarrow{D} N(0,1). \tag{2.8}$$

Let $\theta_\alpha = LD_{100\alpha}$. Applying Taylor's Theorem to the first order, we have

$$\hat{p}(\hat{\theta}_\alpha) = \hat{p}(\theta_\alpha) + \hat{p}'(\theta_\alpha)(\hat{\theta}_\alpha - \theta_\alpha),$$

$$\hat{\theta}_\alpha - \theta_\alpha = -\frac{\hat{p}(\theta_\alpha) - \hat{p}(\hat{\theta}_\alpha)}{\hat{p}'(\theta_\alpha)}. \tag{2.9}$$

We will show that $-\frac{\hat{p}(\theta_\alpha) - \hat{p}(\hat{\theta}_\alpha)}{\hat{p}'(\theta_\alpha)} \rightarrow -\frac{\hat{p}(\theta_\alpha) - p(\theta_\alpha)}{p'(\theta_\alpha)}$. In other words, we will show

$\hat{p}(\hat{\theta}_\alpha) \rightarrow p(\theta_\alpha)$ and $\hat{p}'(\theta_\alpha) \rightarrow p'(\theta_\alpha)$.

First, Müller and Schmitt (1988) show that $\hat{p}(\hat{\theta}_\alpha) \rightarrow p(\theta_\alpha)$. They state the following in Assumption *A(j)*: let $K^{(j)}$ be Lipschitz continuous on the real line and $\lim_{n\to\infty} \log n / nb^{2j+1} = 0$. From Theorem 3 (Müller and Schmitt), assume that $\theta_\alpha \in [\delta, 1-\delta]$ and that Assumption *A(j)* holds for *j* = 0, 1, then $\hat{\theta}_\alpha \rightarrow \theta_\alpha$ a.s. as $n \rightarrow \infty$.

If $\hat{\theta}_\alpha \rightarrow \theta_\alpha$ a.s. as $n \rightarrow \infty$, then from (2.9), $\hat{\theta}_\alpha - \theta_\alpha \rightarrow 0$ and $\hat{p}(\hat{\theta}_\alpha) \rightarrow \hat{p}(\theta_\alpha)$.

Further, by the Weak Law of Large Numbers, $\hat{p}(\theta_\alpha) \rightarrow p(\theta_\alpha)$. Thus, $\hat{p}(\hat{\theta}_\alpha) \rightarrow p(\theta_\alpha)$.

Next, $\hat{p}'(\theta_\alpha) \rightarrow p'(\theta_\alpha)$ follows from Theorem 2 (Müller and Schmitt).

Thus,

$$\hat{\theta}_\alpha - \theta_\alpha = -\frac{\hat{p}(\theta_\alpha) - \hat{p}(\hat{\theta}_\alpha)}{\hat{p}'(\theta_\alpha)} \to -\frac{\hat{p}(\theta_\alpha) - p(\theta_\alpha)}{p'(\theta_\alpha)}.$$

$$E(\hat{\theta}_\alpha - \theta_\alpha) = -\frac{E[\hat{p}(\theta_\alpha) - p(\theta_\alpha)]}{p'(\theta_\alpha)} \quad \text{and} \tag{2.10}$$

$$Var(\hat{\theta}_\alpha - \theta_\alpha) = \frac{Var[\hat{p}(\theta_\alpha) - p(\theta_\alpha)]}{[p'(\theta_\alpha)]^2}. \tag{2.11}$$

Using the results from Equation (2.8), we have

$$E(\hat{\theta}_\alpha - \theta_\alpha) = -\frac{B_{C,n}\sqrt{Var[\hat{p}(\theta_\alpha)]}}{p'(\theta_\alpha)} \tag{2.12}$$

and

$$Var(\hat{\theta}_\alpha - \theta_\alpha) = \frac{Var[\hat{p}(\theta_\alpha)]}{[p'(\theta_\alpha)]^2}, \tag{2.13}$$

where

$$Var[\hat{p}(\theta_\alpha)] = \hat{p}(\theta_\alpha)[1 - \hat{p}(\theta_\alpha)]\frac{1}{h^2}\sum_{i=1}^{n}\left(\int_{s_{i-1}}^{s_i} K\left(\frac{\theta_\alpha - u}{h}\right)du\right)^2. \tag{2.14}$$

Thus, ignoring terms of order 1/*n*,

$$\hat{\theta}_\alpha - \theta_\alpha \sim N\left(-\frac{B_{C,n}\sqrt{Var[\hat{p}(\theta_\alpha)]}}{p'(\theta_\alpha)}, \frac{Var[\hat{p}(\theta_\alpha)]}{[p'(\theta_\alpha)]^2}\right). \tag{2.15}$$

Distributional Properties of Cline Width

Applying the results from Müller and Schmitt, we derive new results for the distribution of the cline width. As defined in the beginning of this Chapter, cline width is the distance between the locations that result in 20% and 80% prevalence, computed as

the difference between LD_{20} and LD_{80}. From the results of Equation (2.15), the distribution of the cline width is

$$(\hat{\theta}_{.8} - \hat{\theta}_{.2}) - (\theta_{.8} - \theta_{.2})$$
$$\sim N\left(\frac{B_{C,n}\sqrt{Var[\hat{p}(\theta_{.2})]}}{p'(\theta_{.2})} - \frac{B_{C,n}\sqrt{Var[\hat{p}(\theta_{.8})]}}{p'(\theta_{.8})}, \frac{Var[\hat{p}(\theta_{.2})]}{[p'(\theta_{.2})]^2} + \frac{Var[\hat{p}(\theta_{.8})]}{[p'(\theta_{.8})]^2} - 2Cov(\hat{\theta}_{.2}, \hat{\theta}_{.8}) \right). \tag{2.16}$$

The computation of $Cov(\hat{\theta}_{.2}, \hat{\theta}_{.8})$ is shown in the next section.

Covariance of $\hat{\theta}_{.2}$ and $\hat{\theta}_{.8}$

The variance of the cline width is computed as

$$Var(\hat{\theta}_{.2}) + Var(\hat{\theta}_{.8}) - 2\, Cov(\hat{\theta}_{.2}, \hat{\theta}_{.8}).$$

The covariance of $\hat{\theta}_{.2}$ and $\hat{\theta}_{.8}$ is derived as follows.

$Cov(\hat{\theta}_{.2}, \hat{\theta}_{.8}) = E((\theta_{.2} - \hat{\theta}_{.2})(\theta_{.8} - \hat{\theta}_{.8}))$. Applying the Mean Value Theorem, there exists a mean value ξ between $\hat{\theta}_{.2}$ and $\theta_{.2}$ such that

$$\hat{p}(\theta_{.2}) = \hat{p}(\hat{\theta}_{.2}) + (\theta_{.2} - \hat{\theta}_{.2})\hat{p}'(\xi)$$

$$\hat{\theta}_{.2} - \theta_{.2} = \frac{\hat{p}(\hat{\theta}_{.2}) - \hat{p}(\theta_{.2})}{\hat{p}'(\xi)} \tag{2.17}$$

$$|\hat{p}'(\xi) - p'(\theta_{.2})| = |\hat{p}'(\xi) - p'(\xi) + p'(\xi) - p'(\theta_{.2})|.$$

By the Triangle Inequality,

$$|\hat{p}'(\xi) - p'(\theta_{.2})| \le |p'(\xi) - \hat{p}'(\xi)| + |p'(\xi) - p'(\theta_{.2})|$$

$$\le \sup_{\xi} |p'(\xi) - \hat{p}'(\xi)| + |p'(\xi) - p'(\theta_{.2})|.$$

As stated in Theorem 2 (Müller and Schmitt), for $j = 1$, $\xi \in [\delta, 1-\delta]$, for some $\delta > 0$,

$$\sup_{\xi \in [\delta, 1-\delta]} | \hat{p}'(\xi) - p'(\xi) | \to 0 \quad \text{a.s} \quad \text{as } n \to \infty.$$

As shown in the earlier section, $\hat{\theta}_{.2} \to \theta_{.2}$ a.s. as $n \to \infty$, and since $\hat{\theta}_{.2} < \xi < \theta_{.2}$,

$\xi \to \theta_{.2}$ as $n \to \infty$ a.s.

By the continuity of p',

$$| p'(\xi) - p'(\theta_{.2}) | \to 0 \quad \text{a.s. as } n \to \infty.$$

Thus,

$$| \hat{p}'(\xi) - p'(\theta_{.2}) | \to 0 \quad \text{a.s. as } n \to \infty. \tag{2.18}$$

From (2.17),

$$\frac{\hat{p}(\hat{\theta}_{.2}) - \hat{p}(\theta_{.2})}{\hat{p}'(\xi)} = \hat{\theta}_{.2} - \theta_{.2}.$$

As shown earlier in the above section, $\hat{p}(\hat{\theta}_{.2}) \to p(\theta_{.2})$ a.s. as $n \to \infty$.

By Slutsky Theorem and from (2.18),

$$-\frac{\hat{p}(\theta_{.2}) - p(\theta_{.2})}{p'(\theta_{.2})} \to \hat{\theta}_{.2} - \theta_{.2}. \tag{2.19}$$

Similarly for LD80, $-\dfrac{\hat{p}(\theta_{.8}) - p(\theta_{.8})}{p'(\theta_{.8})} \to \hat{\theta}_{.8} - \theta_{.8}$.

Thus, to the first order,

$$Cov(\hat{\theta}_{.2}, \hat{\theta}_{.8}) = E((\theta_{.2} - \hat{\theta}_{.2})(\theta_{.8} - \hat{\theta}_{.8})) = E\left[\left(-\frac{p(\theta_{.2}) - \hat{p}(\theta_{.2})}{p'(\theta_{.2})}\right)\left(-\frac{p(\theta_{.8}) - \hat{p}(\theta_{.8})}{p'(\theta_{.8})}\right)\right]$$

$$= \frac{Cov(\hat{p}(\theta_{.2}), \hat{p}(\theta_{.8}))}{p'(\theta_{.2})\, p'(\theta_{.8})}. \tag{2.20}$$

The $Cov(\hat{p}(\theta_{.2}), \hat{p}(\theta_{.8}))$ term can be expressed as follows.

$$Cov(\hat{p}(\theta_{.2}), \hat{p}(\theta_{.8})) = E(\hat{p}(\theta_{.2}), \hat{p}(\theta_{.8})) - E(\hat{p}(\theta_{.2}))E(\hat{p}(\theta_{.8})),$$

where

$$E(\hat{p}(\theta_{.2}), \hat{p}(\theta_{.8})) = \frac{1}{h^2} E\left[\sum_{i=1}^{n} \int_{s_{i-1}}^{s_i} y_i K\left(\frac{\theta_{.2} - u}{h} \right) du \sum_{i=1}^{n} \int_{s_{i-1}}^{s_i} y_i K\left(\frac{\theta_{.8} - u}{h} \right) du \right].$$

$h^2 E(\hat{p}(\theta_{.2}), \hat{p}(\theta_{.8}))$ =

$$E\left[\sum_{i=1}^{n} y_i^{\,2} \int_{s_{i-1}}^{s_i} K\left(\frac{\theta_{.2} - u}{h} \right) du \int_{s_{i-1}}^{s_i} K\left(\frac{\theta_{.8} - u}{h} \right) du + 2 \sum_{i \neq j} \sum y_i y_j \int_{s_{i-1}}^{s_i} K\left(\frac{\theta_{.2} - u}{h} \right) du \int_{s_{j-1}}^{s_j} K\left(\frac{\theta_{.8} - u}{h} \right) du \right]$$

$$= E\left[\sum_{i=1}^{n} (p(x_i)(1 - p(x_i)) + p^2(x_i)) \int_{s_{i-1}}^{s_i} K\left(\frac{\theta_{.2} - u}{h} \right) du \int_{s_{i-1}}^{s_i} K\left(\frac{\theta_{.8} - u}{h} \right) du \right]$$

$$+ 2E\left[\sum_{i \neq j} \sum p(x_i) p(x_j) \int_{s_{i-1}}^{s_i} K\left(\frac{\theta_{.2} - u}{h} \right) du \int_{s_{j-1}}^{s_j} K\left(\frac{\theta_{.8} - u}{h} \right) du \right].$$

Thus, $Cov(\hat{p}(\theta_{.2}), \hat{p}(\theta_{.8}))$ =

$$\frac{1}{h^2} E\left[\sum_{i=1}^{n} (p(x_i)(1 - p(x_i)) + p^2(x_i)) \int_{s_{i-1}}^{s_i} K\left(\frac{\theta_{.2} - u}{h} \right) du \int_{s_{i-1}}^{s_i} K\left(\frac{\theta_{.8} - u}{h} \right) du \right]$$

$$+ 2E\left[\sum_{i \neq j} \sum p(x_i) p(x_j) \int_{s_{i-1}}^{s_i} K\left(\frac{\theta_{.2} - u}{h} \right) du \int_{s_{j-1}}^{s_j} K\left(\frac{\theta_{.8} - u}{h} \right) du \right]$$

$$- \frac{1}{h}\left[\sum_{i=1}^{n} p(x_i) \int_{s_{i-1}}^{s_i} K\left(\frac{\theta_{.2} - u}{h} \right) du \right] \frac{1}{h}\left[\sum_{j=1}^{n} p(x_j) \int_{s_{j-1}}^{s_j} K\left(\frac{\theta_{.8} - u}{h} \right) du \right]. \qquad (2.21)$$

With the computation of $Cov(\hat{p}(\theta_{.2}), \hat{p}(\theta_{.8}))$ above, the covariance of LD_{20} and LD_{80} can be computed as

$$Cov(\hat{\theta}_{.2}, \hat{\theta}_{.8}) = \frac{Cov(\hat{p}(\theta_{.2}), \hat{p}(\theta_{.8}))}{p'(\theta_{.2}) p'(\theta_{.8})}.$$

This is a workable computational form and simulation has shown that this covariance behaves much like the variance of an LD_{20}. Similar to the variance, the covariance is proportional to the bandwidth and inversely proportional to the sample size, the design density, and the curvature of $p(x)$. As we will see in the simulation chapter, this covariance term is generally quite small.

CHAPTER 3

CONFIDENCE INTERVALS

In this chapter we develop several approaches to construct confidence intervals for LD_{20}, LD_{80}, and the cline width. We evaluate seven methods for computing confidence intervals. The first two methods were formulated by Müller and Schmitt (1988). In constructing the confidence intervals, Müller and Schmitt ignored the bias. We extend Müller and Schmitt's results by correcting their confidence intervals for bias and investigating their behavior for small samples. This bias-corrected confidence interval is presented in Method 3. The fourth through seventh methods are based on the distributional results (Hart 1997) shown in Chapter 2.

Method 1: Müller and Schmitt

From Theorem 4 (Müller and Schmitt), Equation (2.10), and Equation (2.11), the distribution of $\hat{\theta}_\alpha$ satisfies

$$(nh)^{1/2}(\hat{\theta}_\alpha - \theta_\alpha) \xrightarrow{D} N\left[\frac{\tau p''(\theta_\alpha)\sigma_K^2}{2p'(\theta_\alpha)}, \frac{\alpha(1-\alpha)J_K}{p'(\theta_\alpha)^2}\right],$$

where $J_K = \int_{-1}^{1} K^2(u)\,du$, and τ has the following properties: $nh^5 \to \tau^2$ as $n \to \infty$ for some τ.

Müller and Schmitt assumed that the bias can be ignored and computed the

confidence intervals for $\hat{\theta}_\alpha$ based on the distribution $N\left[\theta_\alpha, \frac{\alpha(1-\alpha)J_K}{nh\hat{p}'(\theta_\alpha)^2}\right]$. Using Theorems 2 and 3 and Expressions (3.5) and (A.1) (Müller and Schmitt), they showed that a consistent estimator of the variance is

$$v_\alpha = \alpha(1-\alpha)\sum_{i=1}^{n}\left(\frac{\left(\frac{1}{h}\int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right)du\right)^2}{\hat{p}'(\hat{\theta}_\alpha)^2}\right).$$

A 100(1- β)% confidence interval for θ_α is

$$\hat{\theta}_\alpha \pm Z_{\frac{\beta}{2}}\sqrt{v_\alpha}\ .$$

In the variance v_α, the first derivative $\hat{p}'(\hat{\theta}_\alpha)$ was estimated for the kernel

$$\frac{15}{16}\left(1-2x^2+x^4\right).$$

It follows that a 100(1- β)% confidence interval for the cline width is

$$(\hat{\theta}_{.8}-\hat{\theta}_{.2}) \pm Z_{\frac{\beta}{2}}\sqrt{v_{.2}+v_{.8}-2Cov(\hat{\theta}_{.2},\hat{\theta}_{.8})}\ .$$

<u>Method 2: Müller and Schmitt</u>
<u>(Difference Quotient)</u>

Instead of evaluating $\hat{p}'(\hat{\theta}_\alpha)$ in v_α, Müller and Schmitt approximated the derivative by an one-sided difference quotient. Defining

$$\xi_\alpha = \left(\alpha(1-\alpha)\sum_{i=1}^{n}\left(\frac{1}{h}\int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right)du\right)^2\right)^{1/2},$$

Müller and Schmitt obtained difference quotients for the right and left sides of θ_α as $\Delta_r = \xi_\alpha / (\hat{\theta}_{\alpha+\xi_\alpha} - \hat{\theta}_\alpha)$ and $\Delta_l = \xi_\alpha / (\hat{\theta}_\alpha - \hat{\theta}_{\alpha-\xi_\alpha})$, respectively. The resulting variances are $v_r = (\hat{\theta}_{\alpha+\xi_\alpha} - \hat{\theta}_\alpha)^2$ and $v_l = (\hat{\theta}_\alpha - \hat{\theta}_{\alpha-\xi_\alpha})^2$. An asymmetric confidence interval for θ_α is constructed as

$$(\hat{\theta}_\alpha - Z_{\frac{\beta}{2}}(\hat{\theta}_\alpha - \hat{\theta}_{\alpha-\xi_\alpha}), \hat{\theta}_\alpha + Z_{\frac{\beta}{2}}(\hat{\theta}_{\alpha+\xi_\alpha} - \hat{\theta}_\alpha)).$$

Similarly, a lower bound of the asymmetric confidence interval for the cline width is

$$(\hat{\theta}_{.8} - \hat{\theta}_{.2}) - Z_{\frac{\beta}{2}} \sqrt{(\hat{\theta}_{.2} - \hat{\theta}_{.2-\xi_{.2}})^2 + (\hat{\theta}_{.8} - \hat{\theta}_{.8-\xi_{.8}})^2 - 2Cov(\hat{\theta}_{.2}, \hat{\theta}_{.8})}$$

and an upper bound of the asymmetric confidence interval for the cline width is

$$(\hat{\theta}_{.8} - \hat{\theta}_{.2}) + Z_{\frac{\beta}{2}} \sqrt{(\hat{\theta}_{.2+\xi_{.2}} - \hat{\theta}_{.2})^2 + (\hat{\theta}_{.8+\xi_{.8}} - \hat{\theta}_{.8})^2 - 2Cov(\hat{\theta}_{.2}, \hat{\theta}_{.8})}.$$

Method 3: Müller and Schmitt (Bias-Corrected)

Müller and Schmitt assumed that the bias could be neglected. Our simulation study in Chapter 4 has shown that the bias of LD_{20}, LD_{80}, and the cline width cannot be ignored, so we extended the Müller and Schmitt's confidence intervals by adjusting for bias. Using the results from Theorem 4 (Müller and Schmitt), we obtained

$$E((nh)^{1/2}(\hat{\theta}_\alpha - \theta_\alpha)) = \frac{\pi p''(\theta_\alpha)\sigma_K^2}{2p'(\theta_\alpha)}. \quad (3.1)$$

To have a more specific understanding of how bias affects the confidence interval, we assume that *p* is the probit curve with the following distribution form for *p*(*d*):

$$p(d) = \Phi\left(\frac{d-\mu}{\sigma}\right),$$

where Φ is the normal cumulative distribution function, μ is the location, and σ is the slope parameter. The first and second derivatives of $p(d)$ are

$$p'(d) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{d-\mu}{\sigma}\right)^2} \text{ and}$$

$$p''(d) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{d-\mu}{\sigma}\right)^2} \left(-\left(\frac{d-\mu}{\sigma^2}\right)\right), \text{ respectively.}$$

As stated in Method 1, $nh^5 \to \tau^2$ as $n \to \infty$ for some τ. At LD_{20}, the probit curve is concave up with positive $p''(d)$; the bias is positive and we are overestimating $p(\theta_{.2})$. To correct the positive bias shown in Equation (3.1), τ has to be negative. Similarly, at LD_{80}, the probit curve is concave down with negative $p''(d)$; the bias is negative and we are underestimating $p(\theta_{.8})$. To correct this negative bias, τ has to be negative as well. Equation (3.1) can then be expressed as

$$E((nh)^{1/2}(\hat{\theta}_\alpha - \theta_\alpha)) = -n^{1/2}h^{5/2}\frac{\sigma_K^2}{2}\left(-\frac{(\theta-\mu)}{\sigma^2}\right).$$

It follows that

$$E(\hat{\theta}_\alpha) = \theta_\alpha\left(1+\frac{h^2\sigma_K^2}{2\sigma}\right) - \frac{h^2\sigma_K^2\mu}{2\sigma^2}.$$

As $n \to \infty$,

$$\frac{\hat{\theta}_\alpha - \theta_\alpha\left(1+\frac{h^2\sigma_K^2}{2\sigma^2}\right) + \frac{h^2\sigma_K^2\mu}{2\sigma^2}}{\sqrt{\frac{\alpha(1-\alpha)J_K}{nhp'(\theta_\alpha)^2}}} \xrightarrow{D} N(0,1).$$

Consequently, the bias-corrected 100(1- β)% confidence interval for θ_α is

$$\frac{\hat{\theta}_\alpha + \frac{h^2\sigma_K^2}{2\sigma^2}\mu}{1+\frac{h^2\sigma_K^2}{2\sigma^2}} \pm Z_{\frac{\beta}{2}} \frac{\sqrt{\frac{\alpha(1-\alpha)J_K}{nhp'(\theta_\alpha)^2}}}{1+\frac{h^2\sigma_K^2}{2\sigma^2}}$$

and the bias-corrected 100(1- β)% confidence interval for the cline width is

$$\frac{\hat{\theta}_{.8} - \hat{\theta}_{.2}}{1+\frac{h^2\sigma_K^2}{2\sigma^2}} \pm Z_{\frac{\beta}{2}} \frac{\sqrt{\frac{\alpha(1-\alpha)J_K}{nhp'(\theta_{.2})^2} + \frac{\alpha(1-\alpha)J_K}{nhp'(\theta_{.8})^2}}}{1+\frac{h^2\sigma_K^2}{2\sigma^2}} - \sqrt{2Cov(\hat{\theta}_{.2}, \hat{\theta}_{.8})}\,.$$

Method 4: Hart ($B_{C,n}$)

Based on the distributional results for θ_α shown in Equation (2.15),

$$\hat{\theta}_\alpha - \theta_\alpha \sim N\left(-\frac{B_{C,n}\sqrt{Var[\hat{p}(\theta_\alpha)]}}{p'(\theta_\alpha)}, \frac{Var[\hat{p}(\theta_\alpha)]}{[p'(\theta_\alpha)]^2}\right),$$

a 100(1- β)% confidence interval for θ_α is

$$\hat{\theta}_\alpha + \frac{B_{C,n}\sqrt{Var[\hat{p}(\theta_\alpha)]}}{p'(\theta_\alpha)} \pm Z_{\frac{\beta}{2}}\sqrt{\frac{Var[\hat{p}(\theta_\alpha)]}{[p'(\theta_\alpha)]^2}},$$

where $B_{C,n} = \dfrac{C^3 n^{-3/5}\sigma_K^2 p''(x)}{2\sigma\left(\sum_{i=1}^{b}\left\{\int_{s_{i-1}}^{s_i} K\left[n^{1/5}(x-u)/C\right]du\right\}^2\right)^{1/2}}$.

A 100(1- β)% confidence interval for the cline width is

$$\left(\hat{\theta}_{.8}+\frac{B_{C,n}\sqrt{Var[\hat{p}(\theta_{.8})]}}{p'(\theta_{.8})}\right)-\left(\hat{\theta}_{.2}+\frac{B_{C,n}\sqrt{Var[\hat{p}(\theta_{.2})]}}{p'(\theta_{.2})}\right)$$

$$\pm\ Z_{\frac{\beta}{2}}\sqrt{\frac{Var[\hat{p}(\theta_{.2})]}{[p'(\theta_{.2})]^2}+\frac{Var[\hat{p}(\theta_{.8})]}{[p'(\theta_{.8})]^2}-2Cov(\hat{\theta}_{.2},\hat{\theta}_{.8})}$$

Method 5: Hart ($B_{n,h}$)

In place of $B_{c,n}$ in the confidence interval presented in Method 4, we used $B_{n,h}$ shown in Equation (2.7),

$$B_{n,h}=\frac{C^3 n^{-3/5}\sigma_K^2 p''(x)+o(C^2 n^{-2/5})+O(n^{-1})}{2\sigma\left(\sum_{i=1}^{b}\left\{\int_{s_{i-1}}^{s_i}K\left[n^{1/5}(x-u)/C\right]du\right\}^2\right)^{1/2}}.$$

The resulting confidence interval for θ_α is

$$\hat{\theta}_\alpha+\frac{B_{n,h}\sqrt{Var[\hat{p}(\theta_\alpha)]}}{p'(\theta_\alpha)}\pm Z_{\frac{\beta}{2}}\sqrt{\frac{Var[\hat{p}(\theta_\alpha)]}{[p'(\theta_\alpha)]^2}}$$

and the confidence interval for the cline width is

$$\left(\hat{\theta}_{.8}+\frac{B_{n,h}\sqrt{Var[\hat{p}(\theta_{.8})]}}{p'(\theta_{.8})}\right)-\left(\hat{\theta}_{.2}+\frac{B_{n,h}\sqrt{Var[\hat{p}(\theta_{.2})]}}{p'(\theta_{.2})}\right)$$

$$\pm\ Z_{\frac{\beta}{2}}\sqrt{\frac{Var[\hat{p}(\theta_{.2})]}{[p'(\theta_{.2})]^2}+\frac{Var[\hat{p}(\theta_{.8})]}{[p'(\theta_{.8})]^2}-2Cov(\hat{\theta}_{.2},\hat{\theta}_{.8})}.$$

Method 6: Hart (Simulated)

The formula for computing confidence intervals is the same as in Method 5. The only difference is the approach for computing the quantity $B_{n,h}$. In Method 6, we computed $B_{n,h}$ as

$$B_{n,h} = \frac{E[\hat{p}_h(x)] - p(x)}{\sqrt{Var[\hat{p}_h(x)]}},$$

where

$Var(\hat{p}_h(x)) = \frac{\sigma^2}{nh}\frac{1}{f(x)}J_K + O\left(\frac{1}{n}\right)$, $J_K = \int_{-1}^{1} K^2(u)\,du$, $\hat{p}_h(x)$ is estimated from kernel simulations, and the true $p(x)$ is the probit curve.

Method 7: Hart (Asymptotic)

The formula for computing confidence intervals is the same as in Method 5 and $B_{n,h}$ is defined the same manner as Method 6. The only difference is the approach for computing the quantity $B_{n,h}$ asymptotically. In computing $B_{n,h}$, $E[\hat{p}_h(x)]$ and $Var(\hat{p}_h(x))$ are computed as

$$E[\hat{p}_h(x)] = \frac{1}{h}\sum_{i=1}^{n} p(x_i)\int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right)du \text{ and}$$

$$Var(\hat{p}_h(x)) = \frac{\sigma^2}{nh}\frac{1}{f(x)}J_K.$$

CHAPTER 4

SIMULATION

We performed a simulation study to verify the distributional results of LD_{20}, LD_{80}, and the cline width shown in Chapter 2. The simulation study also facilitates the evaluation of confidence intervals for LD_{20}, LD_{80}, and the cline width. We applied each of the seven methods presented in Chapter 3 to compute the corresponding 95% confidence intervals for the estimates.

In our simulation study, we assumed that the true distribution of the location response curve follows the probit distribution with parameters mean, $\mu = 0.5$ and standard deviation, $\sigma = 0.1$. For the probit curve, the values for LD_{20}, LD_{80}, and the cline width are 0.4158, 0.5842, and 0.1684, respectively. This chapter first presents the distributional results of LD_{20}, LD_{80}, and the cline width for the probit curve, followed by the evaluation of confidence intervals for the estimates.

Distributional Results of LD_{20}, LD_{80}, and the Cline Width for the Probit

Mimicking a real data example that we present later in Chapter 5, we created a probit data set with $n = 156$ observations. The distances are equally spaced, resulting in a design density $f(x) = 1$. We examined the distribution of LD_{20}, LD_{80}, and the cline width for the probit. The probit curve is shown in Figure 1.

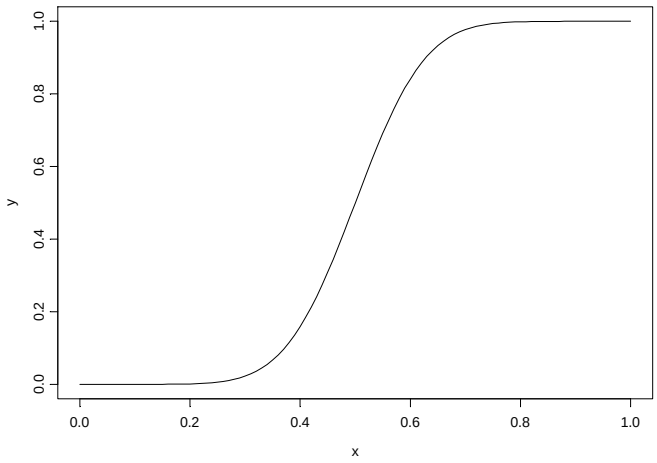

Figure 1. Probit Curve.

For the probit curve shown in Figure 1, $LD_{20} = 0.4158$ and $LD_{80} = 0.5842$. For estimating $\hat{p}(x)$, we determined the value of the bandwidth based on the following formula for finding the optimal local bandwidth: $h = \left(\frac{\alpha(1-\alpha)J_K}{f(x)[p''(x)]^2 \sigma_K^4 n} \right)^{1/5}$, which leads to a bandwidth of 0.123 for this sample size $n = 156$. The relationship between the bandwidth and the sample size is shown in Figure 2. The bandwidth decreases as the sample size increases.

In Equation (2.8), we found that the asymptotic distribution of $\hat{p}(x)$ is

$$\frac{\hat{p}_h(x) - p(x)}{\sqrt{Var[\hat{p}(x)]}} - B_{C,n} \xrightarrow{D} N(0,1).$$

To examine the above distribution, we used the true probit to generate 1,000 replicate samples. For computing $\hat{p}(x)$, we used the optimal bandwidth of 0.123. For $x = LD_{20}$, the

mean and variance of $\frac{\hat{p}_h(x) - p(x)}{\sqrt{Var(\hat{p}(x))}} - B_{c,n}$ are -0.158 and 0.964, respectively. For x = LD_{80}, the mean of $\frac{\hat{p}_h(x) - p(x)}{\sqrt{Var(\hat{p}(x))}} - B_{c,n}$ is 0.148 and the variance is 0.945. We evaluated

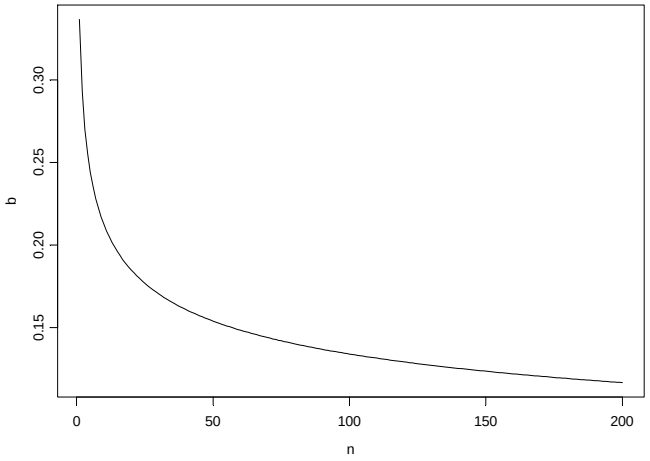

Figure 2. Bandwidth and Sample Size.

normality using Filliben's test which computes a correlation coefficient of a QQ plot (Filliben, 1975). The hypothesis that $\frac{\hat{p}_h(x) - p(x)}{\sqrt{Var(\hat{p}(x))}} - B_{c,n}$ is normally distributed for x = LD_{20}, LD_{80} is rejected (Filliben's correlation is 0.9880 for LD_{20} and 0.9912 for LD_{80}; the approximate 5% cutoff is 0.9982 for a sample size of 1,000; since 0.9880 and 0.9912 are less than 5% cutoff, we reject normality). Due to the small sample size of 156, the means of $\frac{\hat{p}_h(x) - p(x)}{\sqrt{Var(\hat{p}(x))}} - B_{c,n}$ for LD_{20} and LD_{80} deviate from zero. When we increased the

sample size to 600, the means of $\frac{\hat{p}_h(x) - p(x)}{\sqrt{Var(\hat{p}(x))}} - B_{c,n}$ for LD_{20} and LD_{80} are closer to zero (-0.0694 for LD_{20} and 0.0962 for LD_{80}). To develop a simulation of the LD_{20} and $LD_{80,}$ we first begin with $\hat{p}(LD_{20})$ and $\hat{p}(LD_{80})$.

Distribution of $\hat{p}(LD_{20})$ and LD_{20} for the Probit

The distribution of $\hat{p}(LD_{20})$ for the 1,000 replicates samples are shown in the histogram in Figure 3. The mean of simulated $\hat{p}(LD_{20})$ is 0.233 and the variance of simulated $\hat{p}(LD_{20})$ is 0.00489 for the 1,000 replicate samples. The reference point of 0.2 is the true probit value at LD_{20}. $\hat{p}(LD_{20})$ tends to overestimate the probit at LD_{20} due to positive bias when *p(x)* is concave up at LD_{20}. The curve in Figure 3 is the density of the normal distribution with estimated mean 0.236 and estimated variance 0.005003; the mean and the variance are estimated from Equation (2.4).

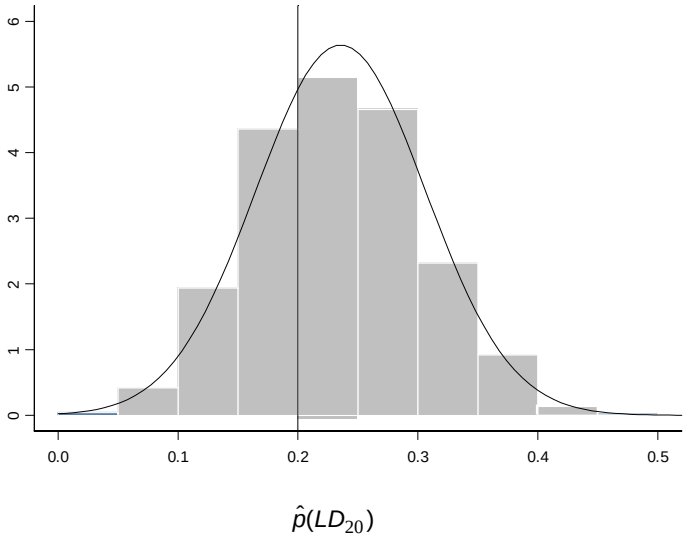

Figure 3. Distribution of $\hat{p}(LD_{20})$.

The distribution of the estimated variance of $\hat{p}(LD_{20})$ using Equation (2.14) for the 1,000 replicate samples is shown in Figure 4. We compared the distribution of the estimated variance with the variance 0.00503 (shown as the vertical line in Figure 4), estimated using Equation (2.14). The estimated variances lie predominantly to the right of 0.00503.

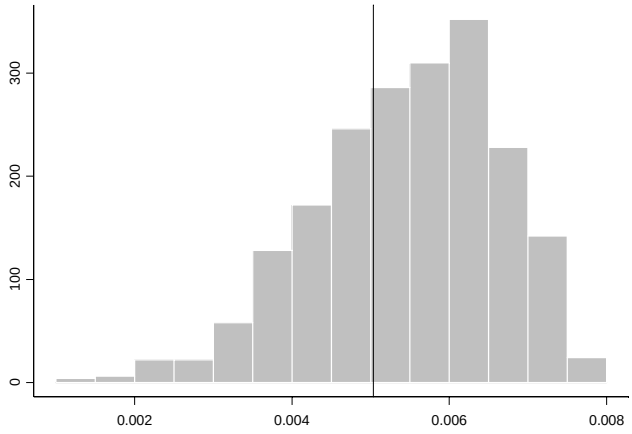

Figure 4. Variance of $\hat{p}(LD_{20})$.

Next, we examined the distribution of $\hat{\theta}_{.2}$. From the results presented in Equation (2.15), theory tells us that the distribution of $\hat{\theta}_{\alpha}$ is given by

$$\hat{\theta}_{\alpha} - \theta_{\alpha} \sim N\left(-\frac{B_{C,n}\sqrt{Var[\hat{p}(\theta_{\alpha})]}}{p'(\theta_{\alpha})}, \frac{Var[\hat{p}(\theta_{\alpha})]}{[p'(\theta_{\alpha})]^2} \right),$$

where $p'(\theta_{\alpha})$ is computed as the first difference of the simulated $\hat{p}(\theta_{\alpha})$ and $Var[\hat{p}(\theta_{\alpha})]$ is computed using Equation (2.14). The distribution of $\hat{\theta}_{.2}$ is normally distributed

(Filliben's correlation 0.9994 for n = 1,000, the approximate 5% cutoff is 0.9982; since 0.9994 > 5% cutoff, we cannot reject normality) as shown in the histogram in Figure 5. For our example the estimated mean and variance of $\hat{\theta}_{.2}$ are 0.4033 and 0.0006383, respectively. The density curve of the normal distribution with mean 0.4033 and variance 0.0006383 is shown in Figure 5. The vertical line shows the true value of $\theta_{.2}$ (0.4158) for the probit curve.

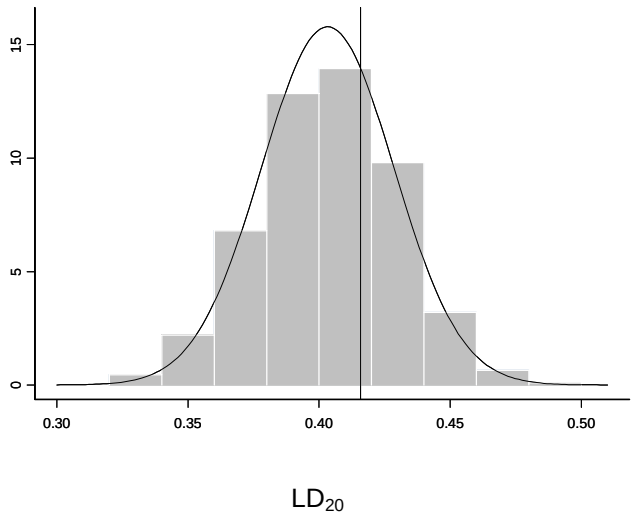

Figure 5. Distribution of $\hat{\theta}_{.2}$.

For the 1,000 replicate samples, the mean of simulated $\hat{\theta}_{.2}$ is 0.4038 and the variance of simulated $\hat{\theta}_{.2}$ is 0.000716. We estimated the mean and the variance of $\hat{\theta}_{.2}$ from Equations (2.12) and (2.13) for the 1,000 replicate samples. The distribution of the estimated mean and the estimated variance are shown in Figure 6 and Figure 7, respectively. The estimated means are mostly distributed to the left of 0.4033 (shown as

the vertical line in Figure 6) and the estimated variances are mostly distributed to the right of 0.000638 (shown as the vertical line in Figure 7).

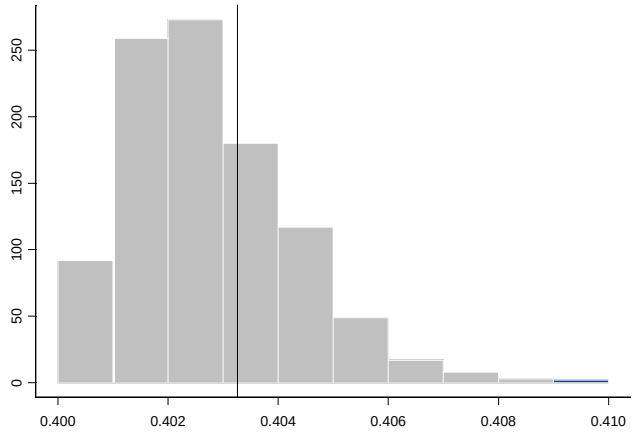

Figure 6. Estimated Mean of $\hat{\theta}_{.2}$.

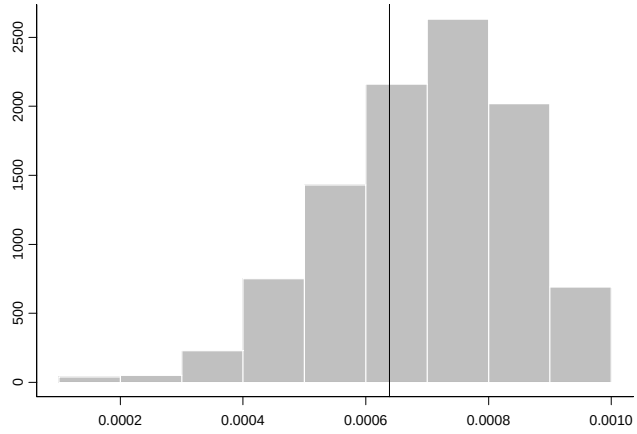

Figure 7. Estimated Variance of $\hat{\theta}_{.2}$.

Distribution of $\hat{p}(LD_{80})$ and LD_{80} for the Probit

The distribution of $\hat{p}(LD_{80})$ for the 1,000 replicates samples are shown in the histogram in Figure 8. The mean of simulated $\hat{p}(LD_{80})$ is 0.766 and the variance of simulated $\hat{p}(LD_{80})$ is 0.00492 for the 1,000 replicate samples. The reference point of 0.8 is the true probit value at LD_{80}. $\hat{p}(LD_{80})$ tends to underestimate the probit at LD_{80}. Figure 8 also shows the density of $\hat{p}(LD_{80})$, estimated from the normal distribution with mean 0.764 and variance 0.005 using Equation (2.4).

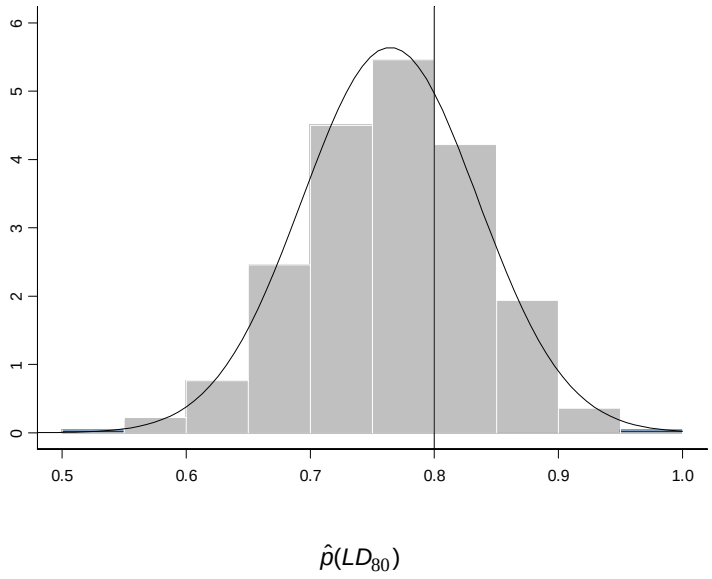

Figure 8. Distribution of $\hat{p}(LD_{80})$.

The distribution of the estimated variance of $\hat{p}(LD_{80})$ using Equation (2.14) for the 1,000 replicate samples is shown in Figure 9. We compared the distribution of the estimated variance with the variance 0.00503 (shown as the vertical line in Figure 9),

estimated using Equation (2.14). The estimated variances lie predominantly to the right of 0.00503.

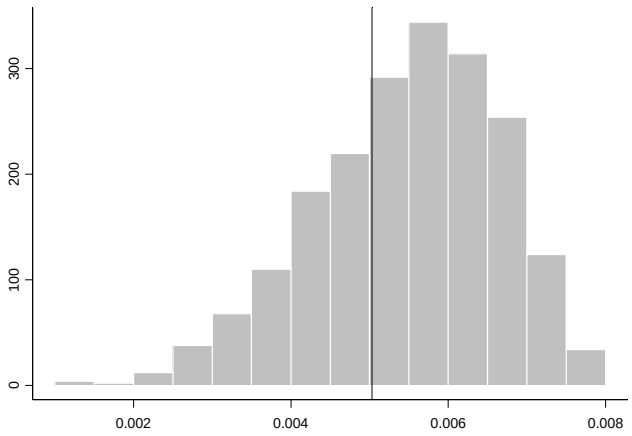

Figure 9. Variance of $\hat{p}(LD_{80})$.

Next, we examined the distribution of $\hat{\theta}_{.8}$. As shown in the histogram in Figure 10, the distribution of $\hat{\theta}_{.8}$ from the simulation is normally distributed (Filliben's correlation is 0.9992 for n = 1,000, the approximate 5% cutoff is 0.9982; since 0.9992 > 5% cutoff, we cannot reject normality). The simulated mean of $\hat{\theta}_{.8}$ is 0.5970 and the simulated variance is 0.0008157. For our example the estimated mean and variance of $\hat{\theta}_{.8}$ are 0.5967 and 0.0006383 respectively. The density of $\hat{\theta}_{.8}$ from a normal distribution with estimated mean 0.5967 and estimated variance 0.0006383 is shown in Figure 10. The vertical line shows the true value of $\theta_{.8}$ (0.5842) for the probit curve.

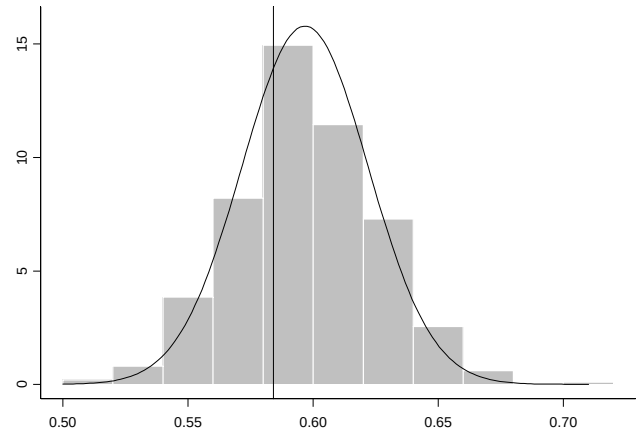

Figure 10. Distribution of $\hat{\theta}_{.8}$.

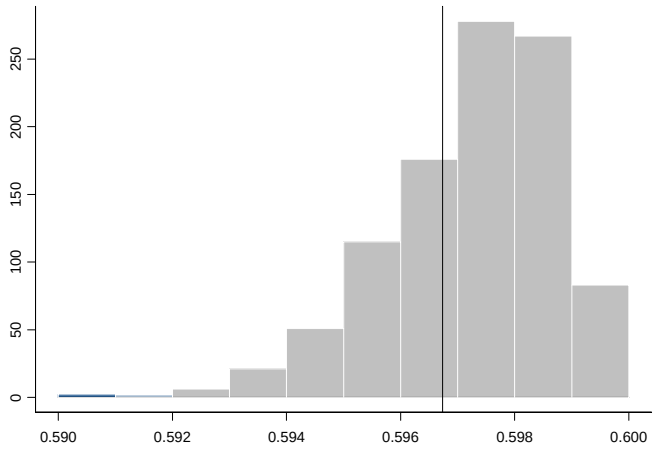

Figure 11. Estimated Mean of $\hat{\theta}_{.8}$.

We also calculated the mean and variance of $\hat{\theta}_{.8}$ for the 1,000 replicate samples using Equations (2.12) and (2.13). The distribution of the estimated mean and the

estimated variance are shown in Figure 11 and Figure 12, respectively. The estimated means are mostly distributed to the right of 0.5967 (shown as the vertical line in Figure 11) and the estimated variances are mostly distributed to the right of 0.000638 (shown as the vertical line in Figure 12).

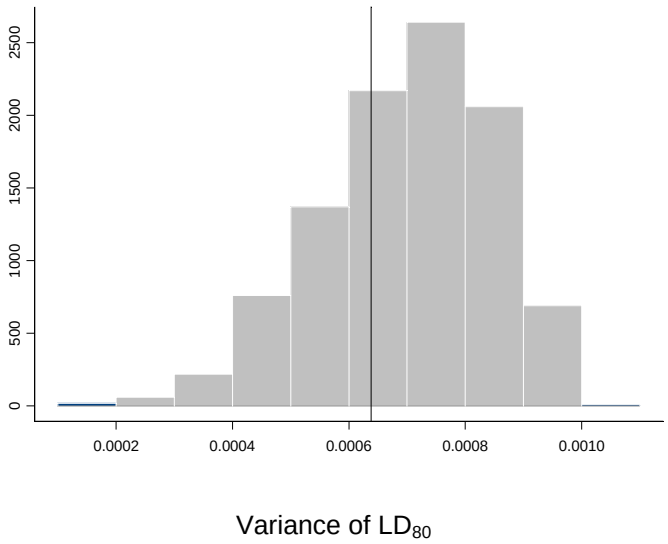

Figure 12. Estimated Variance of $\hat{\theta}_{.8}$.

Distribution of Cline Width for the Probit

The distribution of the cline width $(\hat{\theta}_{.8} - \hat{\theta}_{.2})$ for the 1,000 replicates samples fails a test of normality (Filliben's correlation is 0.9936 for n = 1,000, the approximate 5% cutoff is 0.9982; since 0.9936 < 5% cutoff, we reject normality) as shown in the histogram in Figure 13. The simulated mean of the cline width for the 1,000 replicate samples is 0.1933 and the simulated variance is 0.001346. For our example the estimated mean and variance of the cline width are 0.1935 and 0.001277, respectively, estimated from Equation (2.16) without the covariance term in the calculation of variance. The

density of the cline width from a normal distribution with estimated mean 0.1935 and estimated variance 0.001277 is shown in Figure 13. The vertical line shows the true value of the cline width (0.1684) for the probit curve. Both the estimated and the simulated cline width tend to overestimate the cline width of the probit.

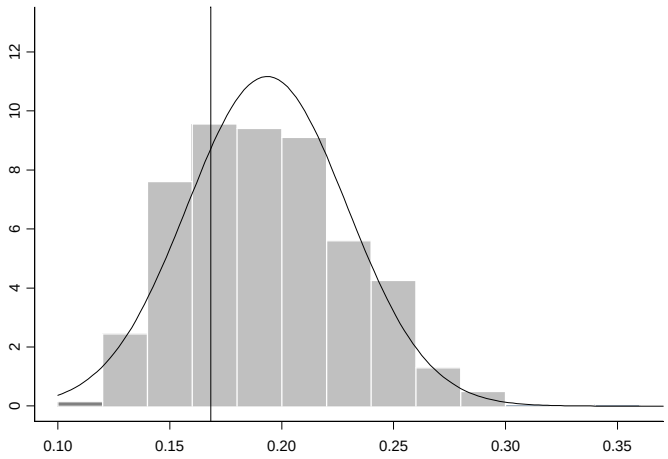

Figure 13. Distribution of the Cline Width.

We also calculated the mean and variance of the cline width for the 1,000 replicate samples using Equation (2.16). The distribution of the estimated mean is shown in Figure 14. The estimated means are mostly distributed to the right of 0.1935 (shown as the vertical line in Figure 14).

We calculated the covariance of $\theta_{.2}$ and $\theta_{.8}$ for the probit data using Equation (2.20). For our example, the estimated covariance is 0.0002965. We calculated the variance of the cline width with and without the covariance term. With the covariance, the estimated variance is 0.000593 less than without.

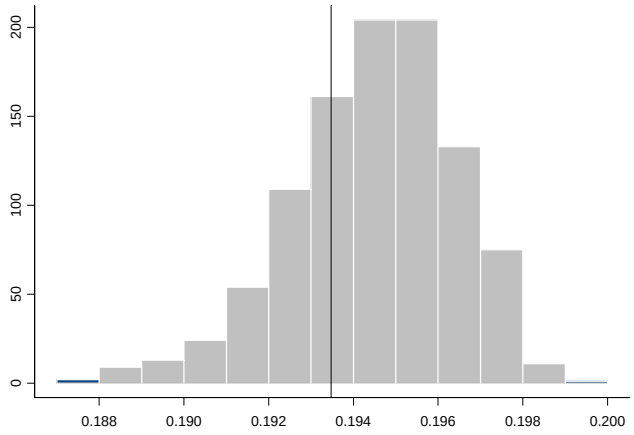

Figure 14. Estimated Mean of $(\hat{\theta}_{.8} - \hat{\theta}_{.2})$.

The distributions of the estimated variance of the cline width with and without the covariance term are shown in Figure 15 and Figure 16, respectively. We compared the estimated variances with 0.0006837 (variance with covariance term) and 0.001277 (variance without covariance term), shown as the vertical lines in the Figures. Both estimated variances (with and without the covariance term) lie predominantly to the right of the reference lines.

In Figure 13, we displayed the distribution of the simulated cline width and observed that the simulated cline width overestimated the true cline width. Using the results shown in Chapter 3, we corrected the cline width for bias and computed the estimated cline width as

$$\left(\hat{\theta}_{.8} + \frac{B_{C,n,.8}\sqrt{Var[\hat{p}(\theta_{.8})]}}{p'(\theta_{.8})}\right) - \left(\hat{\theta}_{.2} + \frac{B_{C,n,.2}\sqrt{Var[\hat{p}(\theta_{.2})]}}{p'(\theta_{.2})}\right).$$

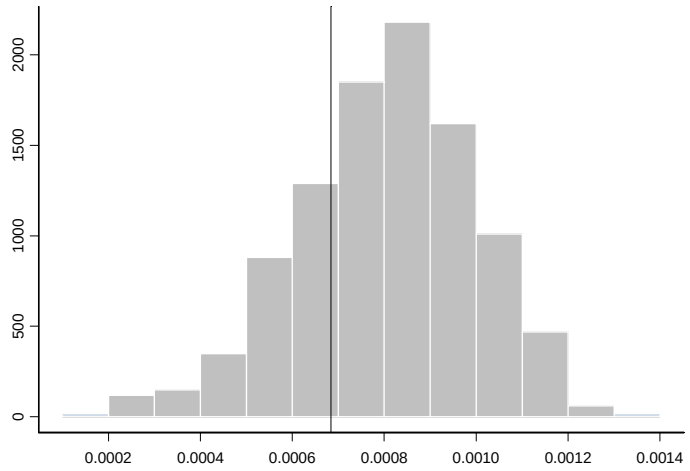

Figure 15. Estimated Variance of $(\hat{\theta}_{.8} - \hat{\theta}_{.2})$ with Covariance Term.

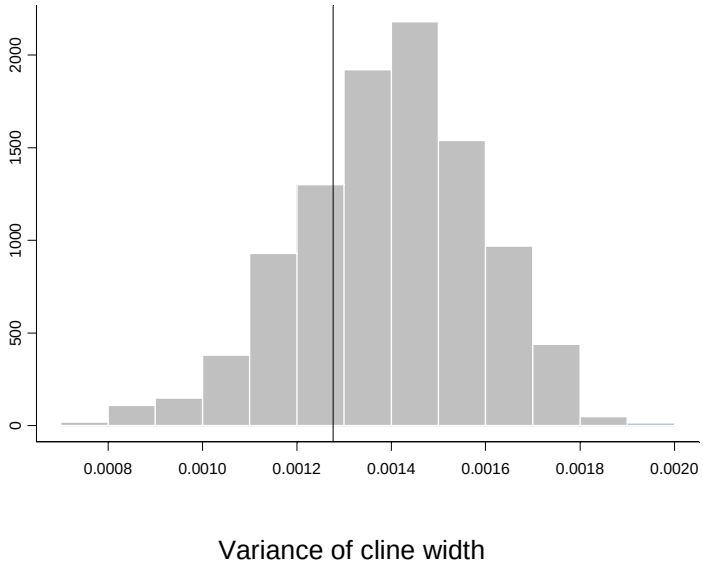

Figure 16. Estimated Variance of $(\hat{\theta}_{.8} - \hat{\theta}_{.2})$ without Covariance Term.

.

In the above formula, the $B_{C,n}$ for LD_{20} is 0.4977, and for LD_{80} the $B_{C,n}$ is -0.4977. The sign of the $B_{C,n}$ agrees with our findings of the simulated LD_{20} and LD_{80}. As seen in Figures 5 and 10, the simulated LD_{20} underestimated the true LD_{20}, whereas the simulated LD_{80} overestimated the true LD_{80}. The distribution of the estimated cline width with bias correction is shown in Figure 17, with a mean of 0.1671 and a variance of 0.00122. The vertical line in Figure 17 is the true cline width. Comparing the simulated cline width in Figure 13 and the estimated cline width in Figure 17, the bias-corrected estimated cline width is a better approximation to the true cline width.

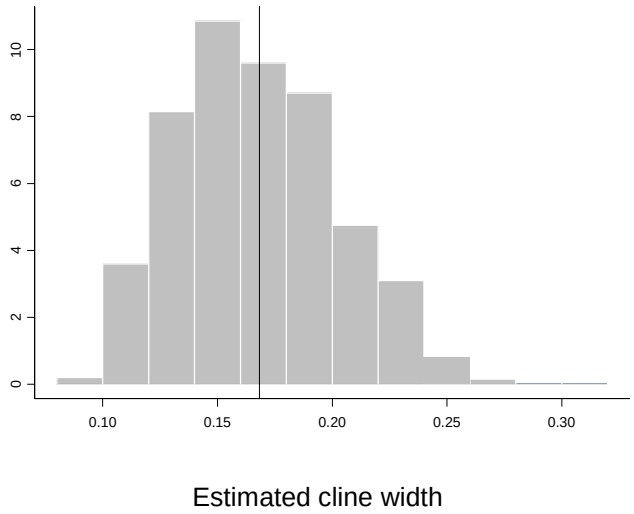

Figure 17. Estimated Cline Width: Adjusting Simulated Cline Width for Bias.

Kernel Estimates and Confidence Intervals for LD_{20}, LD_{80}, and the Cline Width

As shown in Equation (2.2), the variance of the kernel estimator depends on the distribution of the design points. Consequently, the design points also impact the optimal

local bandwidth (Equation (2.6)) and the confidence intervals. To assess the impact of different designs on confidence intervals, we selected equal percentile distances from the uniform distribution and the normal distribution.

We examined the behavior of the kernel estimators for various sample sizes (50, 100, 200, 400, and 800). The kernels that we used in the simulations are optimal or near optimal kernels as referenced by Hart (1997) and Müller and Schmitt (1988). These unimodal kernels are listed as follows and are shown in Figure 18. The kernel *K21* is flatter and less peaked in the center.

K21: $\frac{3}{4}\left(1-u^2\right)$, for -1 $\leq u \leq$ 1

K22: $\frac{15}{16}(1-2u^2+u^4)$, for -1 $\leq u \leq$ 1

K23: $\frac{35}{32}(1-3u^2+3u^4-u^6)$, for -1 $\leq u \leq$ 1.

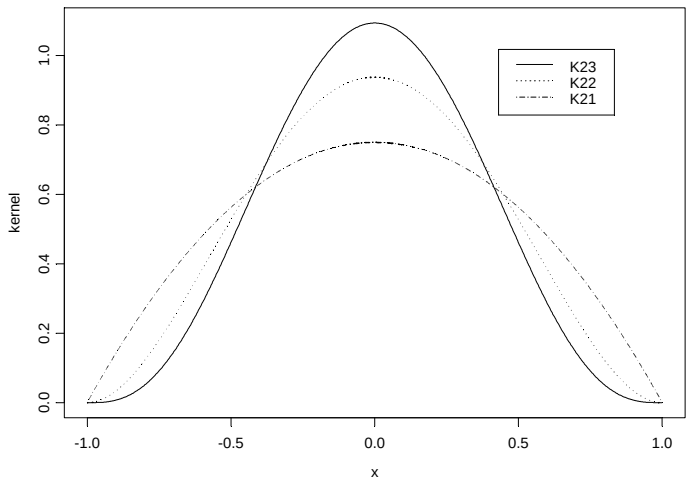

Figure 18. Kernels *K21, K22, and K23*.

For a given sample size, a given kernel, and a given design, we conducted 4,000 simulations. The optimal bandwidths that we computed for the simulation study using Equation (2.6) are shown in Table 1. For the same sample size and design, the bandwidth for *K23* is larger than *K21* and *K22*. The bandwidths for the normal design are smaller than the uniform design within the same sample size and kernel.

Table 1. Optimal Bandwidths by Sample Size, Design, and Kernel

Sample size	Uniform			Normal		
	K21	*K22*	*K23*	*K21*	*K22*	*K23*
50	0.1539	0.1824	0.2071	0.1381	0.1636	0.1858
100	0.1340	0.1588	0.1803	0.1182	0.1401	0.1591
200	0.1167	0.1382	0.1569	0.1015	0.1203	0.1366
400	0.1016	0.1203	0.1366	0.0874	0.1036	0.1176
800	0.0884	0.1047	0.1189	0.0754	0.0893	0.1014

We evaluated the 95% confidence intervals for LD_{20}, LD_{80}, and the cline width based on the mean interval length, the coverage probability of the estimates, and the midpoint of the interval. In our study, we have found that the quality of the kernel estimator does not depend much on the shape of the kernel. The kernels *K21*, *K22*, and *K23* produce similar confidence intervals. The findings presented in the following sections pertain to kernel *K21*, but the results can be generalized to kernels *K22* and *K23*.

Estimation of LD_{20} and LD_{80}

We present the LD_{20} and LD_{80} estimation results for uniform distances first, followed by the results for normal distances.

Uniform Spacing of Distances

Tables 2 through 5 show the mean interval length, the coverage probability of LD_{20}, the interval midpoint, and the variance of LD_{20} using kernel *K21* when the distances are uniformly spaced. The attributes of confidence intervals for LD_{20} shown in Tables 2 through 5 using kernel *K21* with uniform distances are presented graphically in Figure 19.

Table 2. LD_{20}: Mean Interval Length (Kernel *K21*, Uniform Distances)

Method	Sample size				
	50	100	200	400	800
M & S	0.1577	0.1190	0.0900	0.0681	0.0516
M & S, Difference Quotient (DQ)	0.2025	0.1425	0.1035	0.0756	0.0559
M & S, bias corrected	0.2049	0.1444	0.1040	0.0759	0.0560
Hart, *Bnh* simulated	0.1564	0.1185	0.0898	0.0681	0.0516
Hart, *Bnh* asymptotic	0.1564	0.1185	0.0898	0.0681	0.0516
Hart, *Bnh*	0.1564	0.1185	0.0898	0.0681	0.0516
Hart, *Bcn*	0.1564	0.1185	0.0898	0.0681	0.0516

The mean interval length, coverage probability, interval midpoint, and variance for LD_{80} are shown in Figure 20. These characteristics of confidence intervals for LD_{80}

are very similar to those for LD_{20}. The discussions below apply to the results for both LD_{20} and LD_{80}, unless stated otherwise.

Table 3. LD_{20}: Coverage Probability (Kernel *K21*, Uniform Distances)

Method	Sample size				
	50	100	200	400	800
M & S	0.9108	0.8990	0.9043	0.9123	0.9098
M & S, Difference Quotient (DQ)	0.9348	0.9185	0.9213	0.9240	0.9178
M & S, bias corrected	0.9295	0.9220	0.9298	0.9350	0.9343
Hart, *Bnh* simulated	0.9368	0.9238	0.9303	0.9353	0.9340
Hart, *Bnh* asymptotic	0.9368	0.9235	0.9300	0.9350	0.9335
Hart, *Bnh*	0.9293	0.9210	0.9278	0.9350	0.9315
Hart, *Bcn*	0.9298	0.9218	0.9298	0.9350	0.9343

For a given sample size, the mean interval lengths for all Hart methods are identical. The mean interval length for the M & S (bias-corrected) method is slightly shorter than the Hart methods. When the sample size is 100, the mean interval length for the M & S (bias-corrected) method is 0.1 and for the Hart *Bcn* the length is 0.12. When the sample size is 800, all methods including the biased M & S are about 0.05.

Coverage probability is the percentage of replications in which the calculated 95% confidence intervals include the true LD value. When the locations are uniformly spaced, the coverage probabilities are about 93% for Hart methods and the M & S (bias-corrected) method, and slightly lower for the biased M & S methods.

Table 4. LD_{20}: Interval Midpoint (Kernel *K21*, Uniform Distances)

Method	Sample size				
	50	100	200	400	800
M & S	0.3991	0.4020	0.4050	0.4077	0.4094
M & S, Differene Quotient (DQ)	0.3803	0.3928	0.4000	0.4051	0.4078
M & S, bias corrected	0.4185	0.4170	0.4164	0.4164	0.4159
Hart, *Bnh* simulated	0.4156	0.4155	0.4152	0.4156	0.4158
Hart, *Bnh* asymptotic	0.4159	0.4152	0.4154	0.4157	0.4155
Hart, *Bnh*	0.4101	0.4111	0.4124	0.4137	0.4141
Hart, *Bcn*	0.4189	0.4171	0.4164	0.4164	0.4159

Table 5. LD_{20}: Variance (Kernel *K21*, Uniform Distances)

Method	Sample size				
	50	100	200	400	800
M & S	0.00162	0.00092	0.00053	0.00030	0.00017
M & S, Differene Quotient (lower bound)	0.00484	0.00201	0.00095	0.00047	0.00024
M & S, Difference Quotient (upper bound)	0.00211	0.00114	0.00063	0.00035	0.00019
M & S, bias corrected	0.00104	0.00066	0.00041	0.00025	0.00015
Hart, *Bnh* simulated	0.00159	0.00091	0.00052	0.00030	0.00017
Hart, *Bnh* asymptotic	0.00159	0.00091	0.00052	0.00030	0.00017
Hart, *Bnh*	0.00159	0.00091	0.00052	0.00030	0.00017
Hart, *Bcn*	0.00159	0.00091	0.00052	0.00030	0.00017

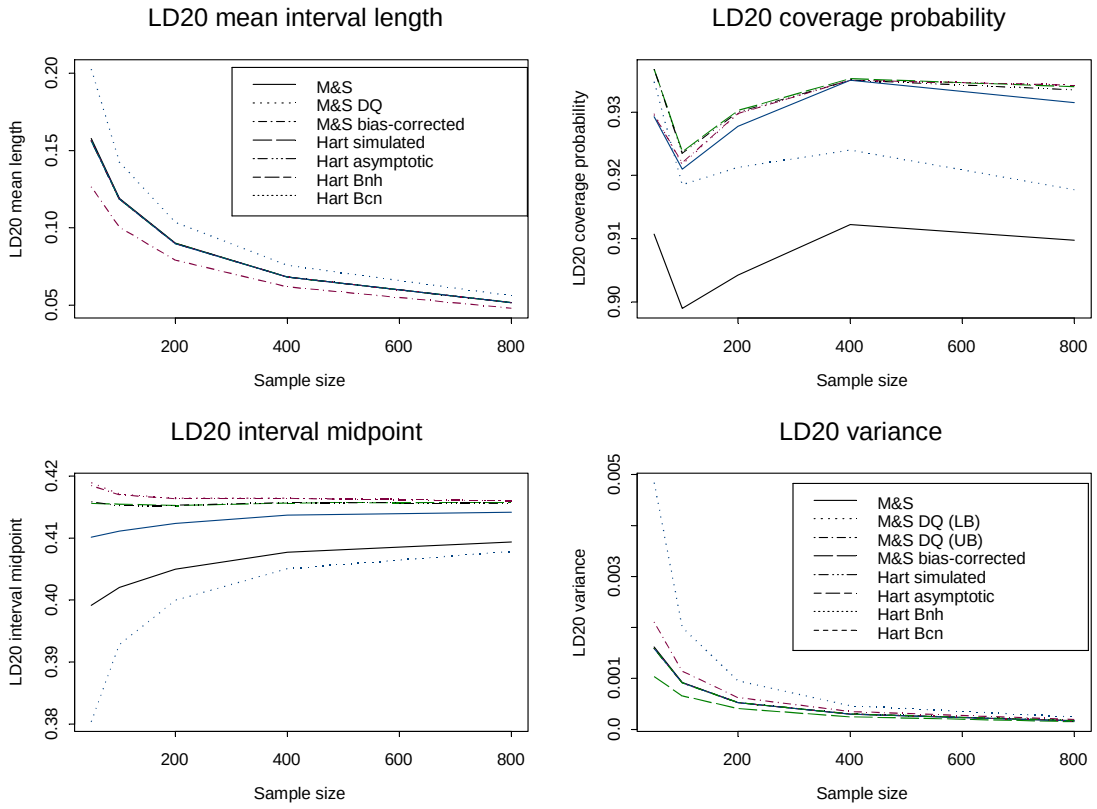

Figure 19. Confidence Intervals and Variance for LD_{20}, Kernel = *K21*, Uniform Spacing of Distances.

The interval midpoint serves as a measure of how close the estimates are to the true values. Hart methods and the M & S (bias-corrected) method provide far better LD estimates that are closer to the true LD values (0.4158 for LD_{20} and 0.5842 for LD_{80}) than the biased M & S methods. The Hart simulated and the Hart asymptotic methods generate LD estimates that are closest to the true LD values, with very small underestimation of LD_{20} and slight overestimation of LD_{80} by the Hart simulated method. For the uniform distances, the *Bnh* method slightly underestimates LD_{20} and overestimates LD_{80}, whereas the *Bcn* method and the M & S (bias-corrected) method slightly overestimate LD_{20} and underestimate LD_{80}. For example, when the sample size is 100, the *Bnh* method

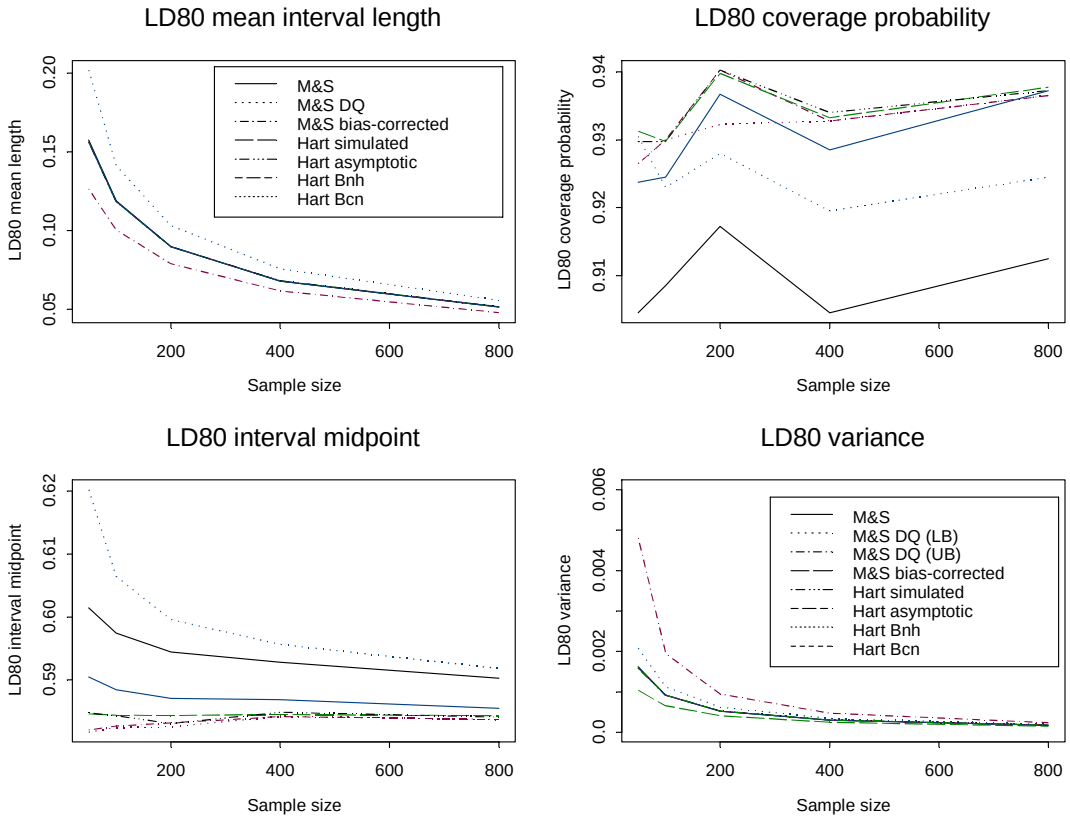

Figure 20. Confidence Intervals and Variance for LD_{80}, Kernel = *K21*, Uniform Spacing of Distances.

underestimates LD_{20} by about 1.1% and the *Bcn* and the M & S (bias-corrected) method overestimate LD_{20} by about 0.3%. The biased M & S methods underestimate LD_{20} and overestimate LD_{80} by a larger amount than the bias correction methods. All methods approach closer to the true LD values as sample size increases.

The variances of LD_{20} are small. For a given sample size, the variances produced by the Hart methods are the same, ranging from 0.0016 for a sample size of 50 to 0.00017

for a sample size of 800. The M & S (bias-corrected) method produces slightly smaller variances than the Hart methods.

Normal Spacing of Distances

Figure 21 and Figure 22 show the mean interval length, the coverage probability, the interval midpoint, and the variance of LD_{20} and LD_{80} using kernel *K21* when the distances are normally spaced. The graphs for LD_{20} and LD_{80} are very similar.

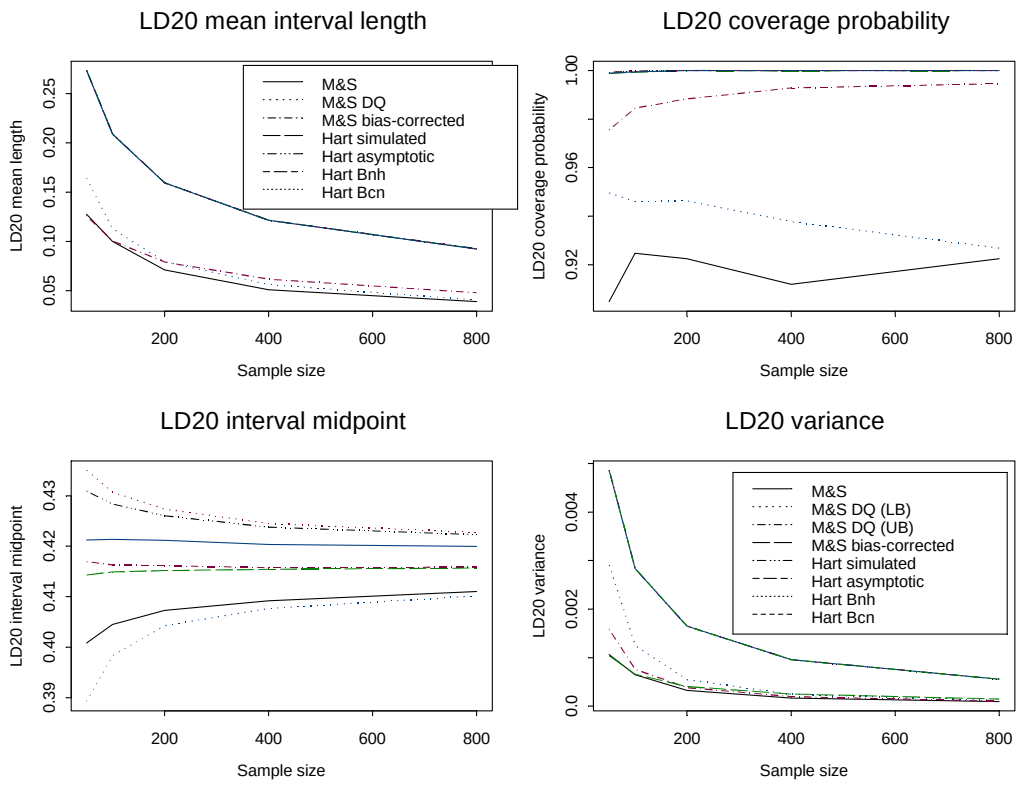

Figure 21. Confidence Intervals and Variance for LD_{20}, Kernel = *K21*, Normal Spacing of Distances.

In the case of the normal spacing of distances, the M & S methods produce shorter mean interval length than the Hart methods. For example, the mean interval

length for the M & S (bias-corrected) method is about half the length of the Hart methods. In general, the coverage probability for all methods is higher than 90%. The Hart methods and the M & S (bias-corrected) method yield very high coverage probability of LD_{20} that are close to 100%. The biased M & S methods produce coverage probability in the low 90%.

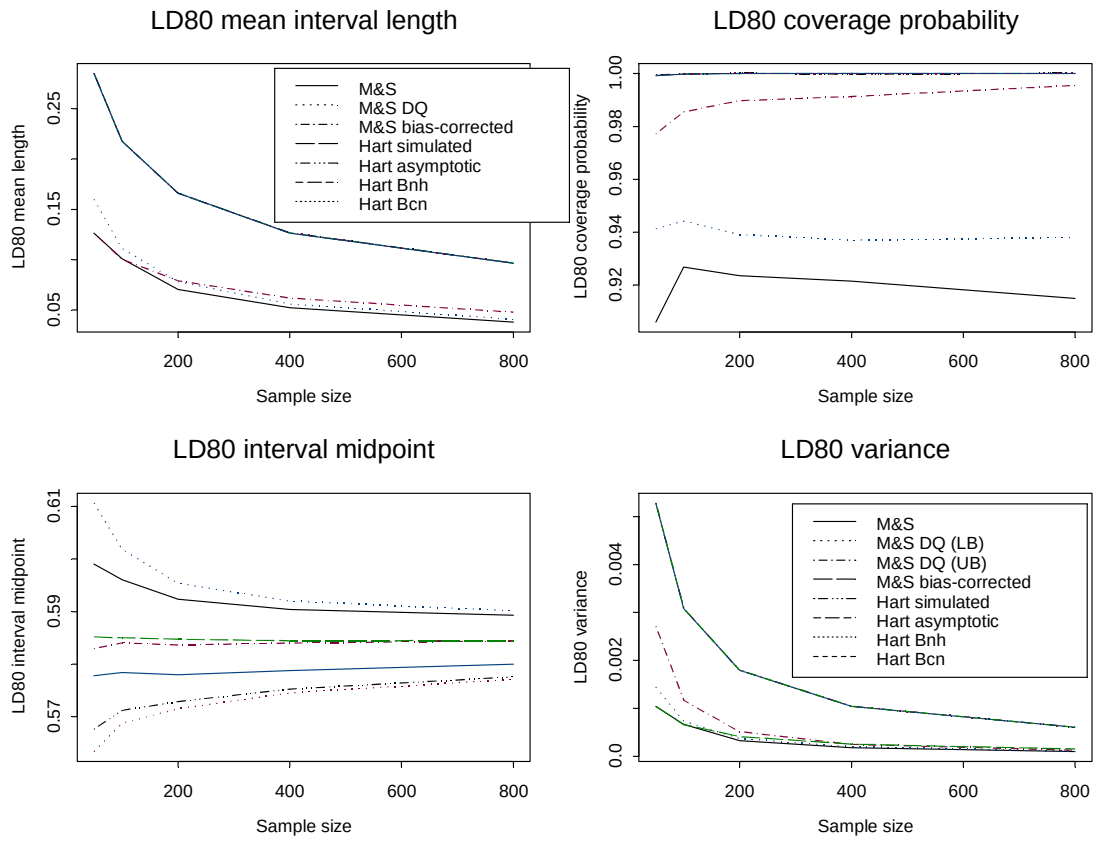

Figure 22. Confidence Intervals and Variance for LD_{80}, Kernel = *K21*, Normal Spacing of Distances.

The M & S (bias-corrected) method provides better LD estimates that are closer to the true LD than the Hart methods. The Hart simulated method generates LD estimates

that are closest to the true LD values. Both the *Bnh* and the *Bcn* methods overestimate LD_{20} and underestimate $LD_{80.}$ However, *Bnh* estimates are closer to the true LD than *Bcn* estimates. For a sample size of 100, the *Bnh* overestimates LD_{20} by 1.3% whereas the *Bcn* overestimates LD_{20} by 3.6%. The extent of underestimating LD_{20} and overestimating LD_{80} is more pronounced in the biased M & S methods than the bias-correction methods. All methods approach closer to the true LD as sample size increases.

For a normal density design, all M & S and Hart methods produce very small LD variances, but they are still approximately three times larger than those for a uniform design density. For a given sample size, the variances produced by the Hart methods are the same, ranging from 0.003 for a sample size of 50 to 0.0006 for a sample size of 800. The variances produced by the M & S (bias-corrected) method are about one-fourth the size of variances produced by the Hart methods.

Estimation of Cline Width

The cline width is the difference between the LD_{80} value and the LD_{20} value, resulting in a true cline width of 0.1684 in our simulation study for the probit curve. We calculate the variance of the cline width in two ways, with and without a covariance term. Both ways yield identical interval midpoint. Including the covariance term lowers the variance of the cline width. For example, the variance of the cline width is 2.36% smaller when the sample size is 100 (0.00592 without covariance term versus 0.00578 with covariance term) using the Hart methods. As a result of smaller variance of the cline width, the method with the covariance term yields shorter interval length and lower

coverage probability. However, the difference becomes miniscule when the sample size is large. In fact, the variances are almost the same when the sample size is 400.

Uniform Spacing of Distances

The attributes of confidence intervals with and without the covariance terms in estimating the variance of the cline width are shown in Figures 23 and 24, respectively. Due to the similarity of the two figures, in the following sections, we discuss the confidence intervals that take into account the covariance in computing the variance of the cline width.

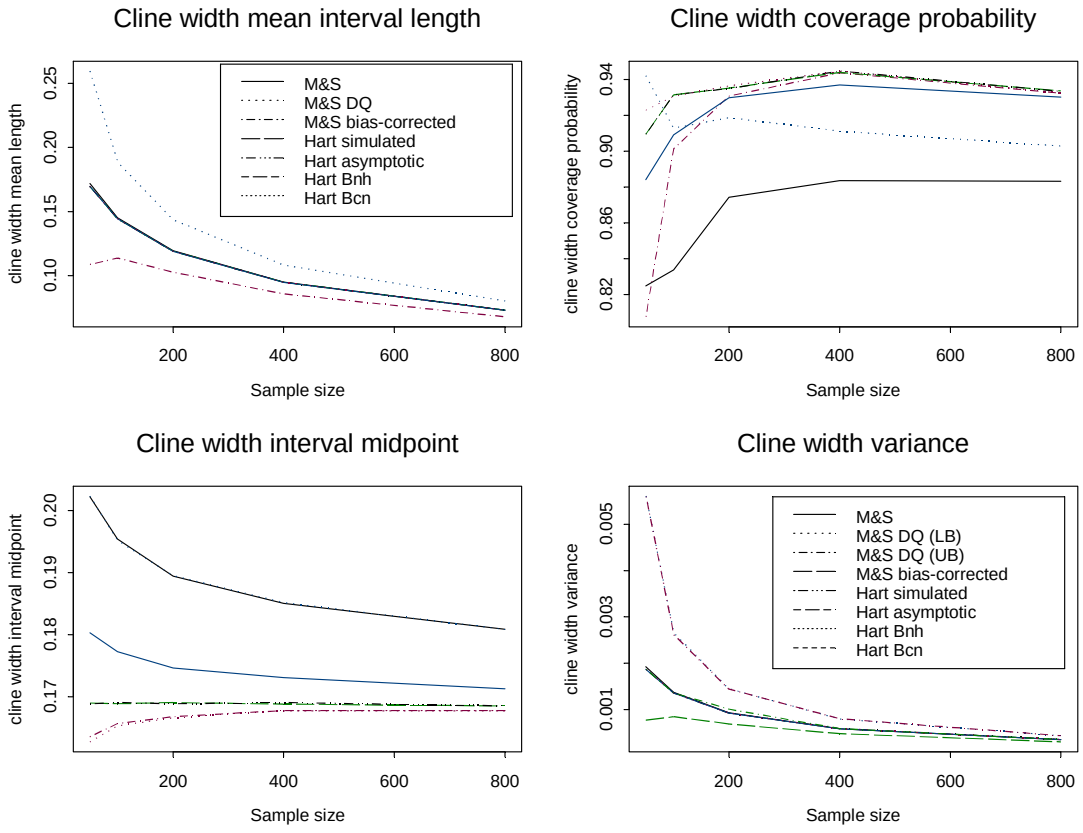

Figure 23. Confidence Intervals and Variance for Cline Width, Kernel = *K21*, Uniform Spacing of Distances (Variance of Cline Width is Computed with the Covariance Term).

For a given sample size, the mean interval lengths for all Hart methods are identical. The interval lengths are shorter for the M & S (bias-corrected) method than the Hart methods. The biased M & S methods produce the longest mean interval length. The Hart methods produce higher coverage probability than the M & S (bias-corrected) method. Coverage probability for Hart methods is above 93%.

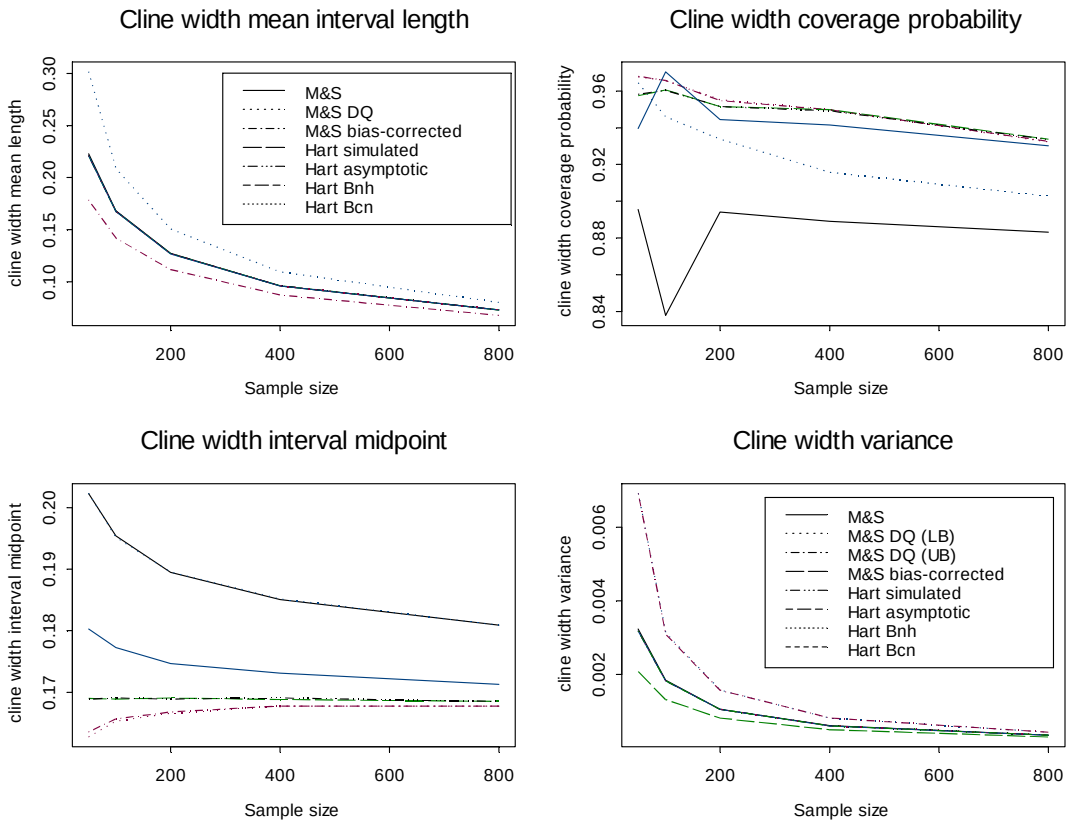

Figure 24. Confidence Intervals and Variance for Cline Width, Kernel = *K21*, Uniform Spacing of Distances (Variance of Cline Width is Computed without the Covariance Term).

Hart methods and the M & S (bias-corrected) method all provide accurate cline width estimates that are close to the true cline width of 0.1684. The Hart simulated and

the Hart asymptotic methods generate cline width estimates that are closest to the true cline width. For the uniform distance, the *Bnh* method overestimates the cline width, whereas the *Bcn* method and the M & S (bias-corrected) method slightly underestimate the cline width. When the sample size is 100, the *Bnh* overestimates the cline width by 5.3% whereas the *Bcn* underestimates the cline width by 1.8%. The biased M & S methods overestimate the cline width, by 15.8% when the sample size is 100. All methods approach closer to the true cline width as sample size increases.

For a given sample size, the variances produced by the Hart methods are the same, ranging from 0.0019 for a sample size of 50 to 0.00035 for a sample size of 800. The variance of the cline width for the M & S (bias-corrected) method is smaller than the Hart methods, about 37% smaller when the sample size is 100.

Normal Spacing of Distances

Figure 25 and Figure 26 show the mean interval length, the coverage probability of the cline width, the interval midpoint, and the variance of the cline width using kernel *K21* when the distances are normally spaced, with and without the covariance term in computing the variance of the cline width.

The M & S (bias-corrected) method produces shorter mean interval length than the Hart methods, about half the length of the Hart methods. For example when the sample size is 100, the mean interval length for the M & S method is 0.13 compared to 0.3 for the Hart methods.

All Hart methods and the M & S (bias-corrected) method produce coverage probability close to 100%. The M & S (bias-corrected) method provides much better

cline width estimates that are very close to the true cline width of 0.1684 than the Hart methods. Both *Bnh* and *Bcn* methods underestimate the cline width. However, *Bnh* estimates are closer to the true cline width than *Bcn* estimates. With a sample size of 100, the underestimation of the cline width is 6.7% and 18% for the *Bnh* and *Bcn*, respectively.

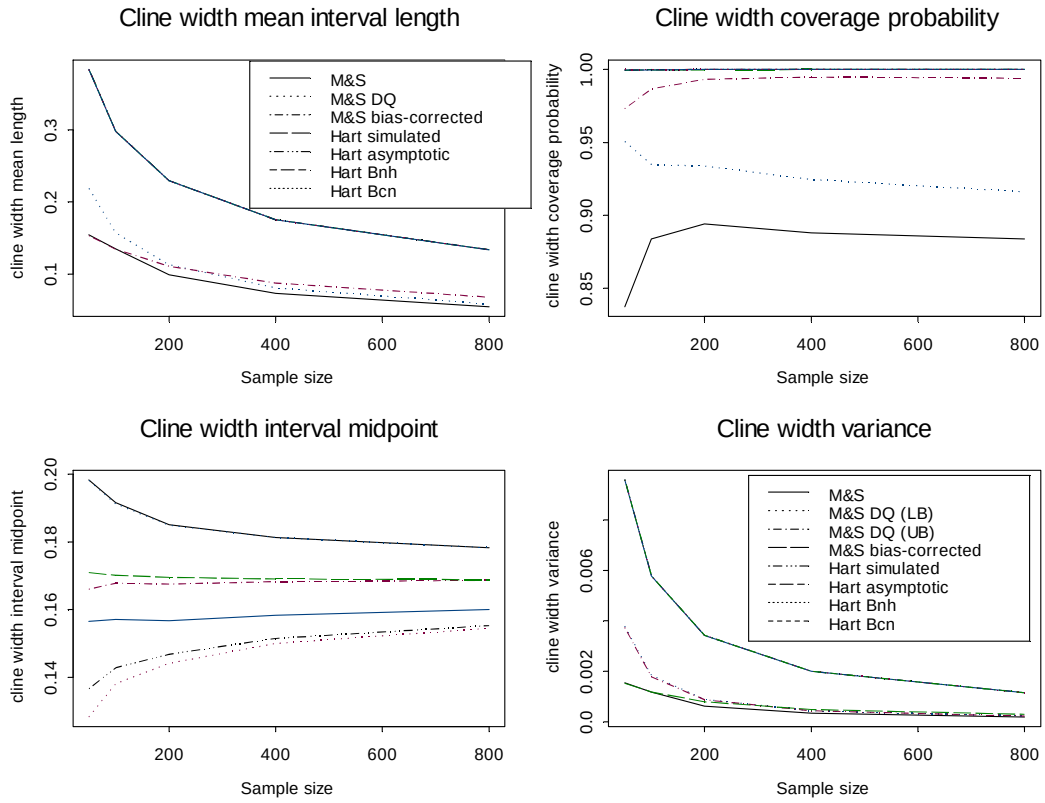

Figure 25. Confidence Intervals and Variance for Cline Width, Kernel = *K21*, Normal Spacing of Distances (Variance of Cline Width is Computed with the Covariance Term).

The M & S (bias-corrected) method yields smaller variances than the Hart methods. For a sample size of 100, the variances of the cline width are 0.001 and 0.0058 for the M & S (bias-corrected) method and the Hart methods, respectively. The variances

produced by the Hart methods are the same for a given sample size, ranging from 0.0096 for a sample size of 50 to 0.0012 for a sample size of 800. These variances are about four times larger than those from the uniformly spaced distance.

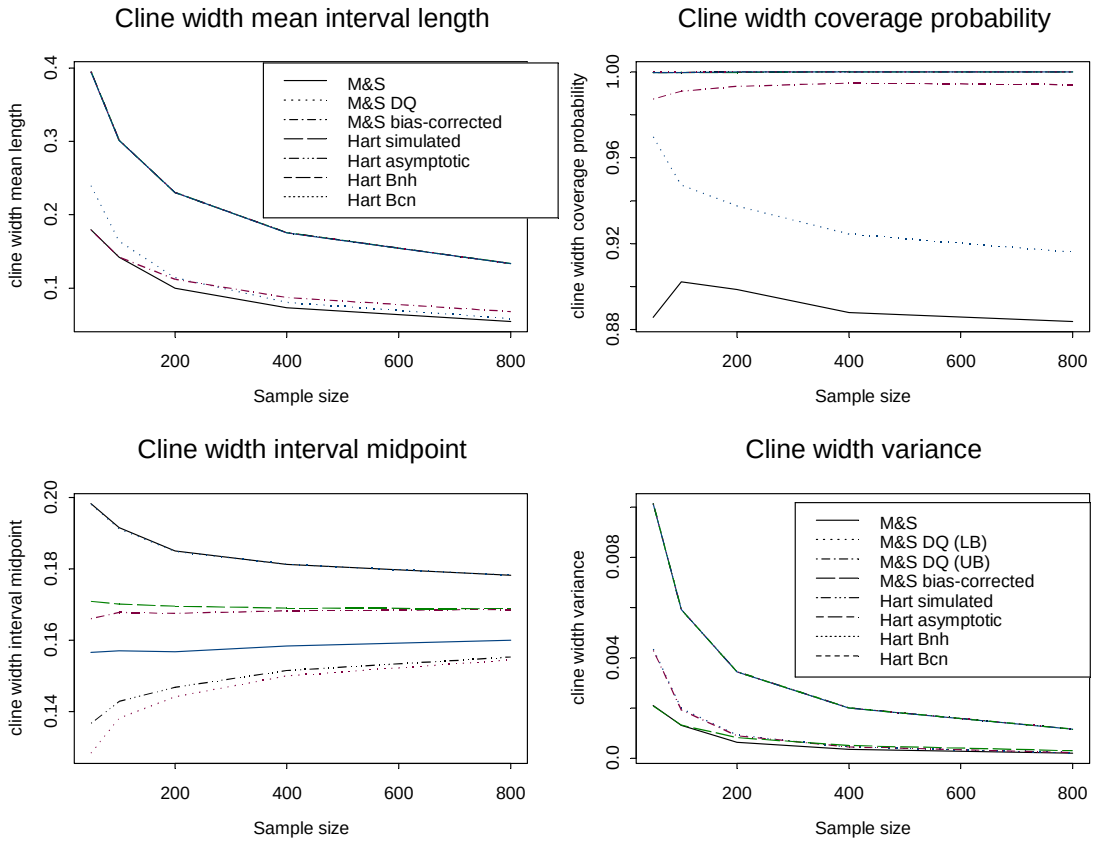

Figure 26. Confidence Intervals and Variance for Cline Width, Kernel = *K21*, Normal Spacing of Distances (Variance of Cline Width is Computed without the Covariance Term).

The performance of the biased M & S methods is poor compared to the bias-corrected methods. The biased M & S methods produce longer interval length, lower coverage probability, and overestimate the cline width by 16% for a sample size of 100.

In summary, *K21*, *K22*, and *K23* kernels generate similar estimates of LD_{20}, LD_{80}, and cline width and confidence intervals. The Hart methods and the M & S (bias-corrected) method outperform the biased M & S methods. These bias corrected methods generate higher coverage probability, shorter mean interval length, and estimates closer to the truth values. As the sample size increases from 50 to 800, we observe the following general results: (1) the mean interval length decreases (for example, with the kernel *K21* and the uniform spacing of distances using the Hart methods, the mean interval length decreases from 0.16 to 0.05 in LD_{20} and LD_{80}, and from 0.17 to 0.07 in the cline width); (2) the estimates approach closer to the true value (for example, with the kernel *K21* and the uniform spacing of distances using the M & S (bias-corrected) method, the cline width estimate is 0.166 with a sample size of 50 compared to 0.1686 with a sample size of 800) and approaches closer to the true cline width value of 0.1684; (3) variance decreases at a rate of $1/n$ as sample size increases; (4) bandwidth decreases as sample size increases. When the sample size doubles, the bandwidth decreases by about 13% and 14% for the uniform and the normal spacing of distances, respectively.

CHAPTER 5

INTERVAL ESTIMATES FOR GENETIC CLINE

In this chapter, we present the interval estimates of LD_{20}, LD_{80}, and the cline width for the mitochondrial DNA (mtDNA) in the Lund (Lu) population of the field vole (Jaarola, 1997). The mtDNA is a genetic marker that characterizes one trait of the field vole (*Microtus agrestis*) population. In a survey conducted from August to October in 1986 to 1992 in southern Sweden, 156 field voles from 36 localities were collected for the mtDNA analysis. This sample size is typical in evolutionary genetics. Samples of mtDNA genetic materials were collected at various distances from a fixed location often a reference population or a geographic landmark. The distances were not equally spaced. The distances, the possession of some allele (number of successes) and the lack of that allele (number of failures) in the Lu population are shown in Table 6.

In our discussion, we will refer to the mtDNA in the Lund population as *lumt*. An estimate of the probability density function of *lumt* distances scaled to [0, 1] using the S-PLUS function *density* is shown in Figure 27. The *density* function in S-PLUS is a smoothing operation that returns the *x* and *y* coordinates of a non-parametric estimate of the probability density of the data. These are kernel estimates. For each *x* value, the window is centered on that *x* and the heights of the window at each data point are summed. After normalization, this sum is the corresponding *y* value. The distances scaled to [0, 1] are shown as dots in Figure 27.

Table 6. mtDNA in Lund Population Data

Distance (km)	No. of successes	No. of failures
0.0	2	0
17.4	17	0
35.4	1	0
39.9	7	0
51.0	6	0
58.2	4	0
67.5	6	1
71.1	1	1
83.5	3	1
83.4	4	3
90.0	1	2
103.5	5	3
105.0	1	1
108.6	1	3
109.8	0	3
112.5	0	4
113.4	0	8
115.8	0	2
116.4	0	3
120.0	0	5
124.5	0	4
132.6	0	3
138.8	0	2
140.7	0	2
141.9	0	3
145.8	0	1
154.2	0	9
155.2	0	7

Table 6 continued

Distance (km)	No. of successes	No. of failures
156.6	0	1
157.9	0	7
159.9	0	2
161.7	0	6
171.0	0	9
175.0	0	1

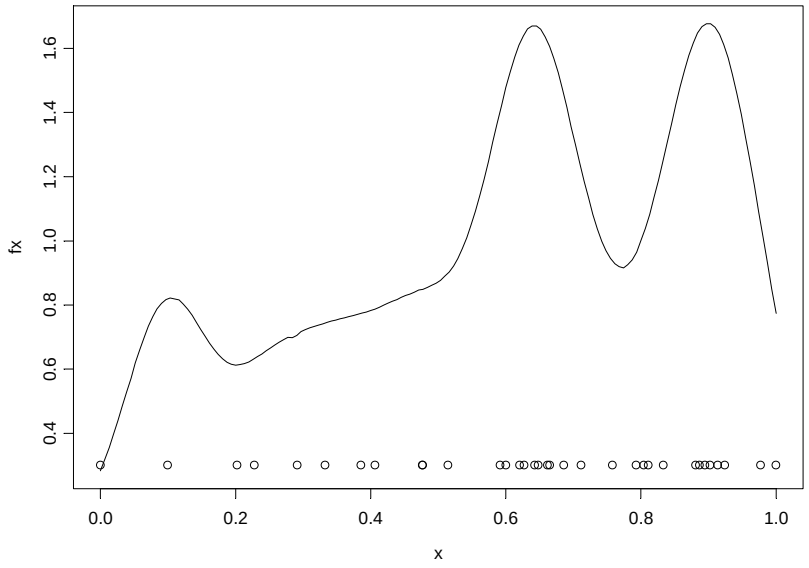

Figure 27. Probability Density Function of *lumt* Distances.

The Gasser-Müller estimator for *lumt* distances with Epanechnikov kernel (i.e., $K(u) = .75(1-u^2)$, $-1 \leq u \leq 1$) and a bandwidth of 0.15 is shown in Figure 28. The kernel estimator, $\hat{p}(x)$ displays a S-shaped curve that is decreasing, starting at a fixed value of $x = 0$ and dropping to a fixed value of $x = 1$. The LD_{20}, LD_{80}, and the cline width are estimated to be 0.656, 0.352, and 0.304, respectively. The number of successes and the

number of failures at each distance (scaled to [0, 1]) are jittered and plotted in Figure 28. The theory we have presented shows that this cline, its LD_{20}, LD_{80}, and width are biased.

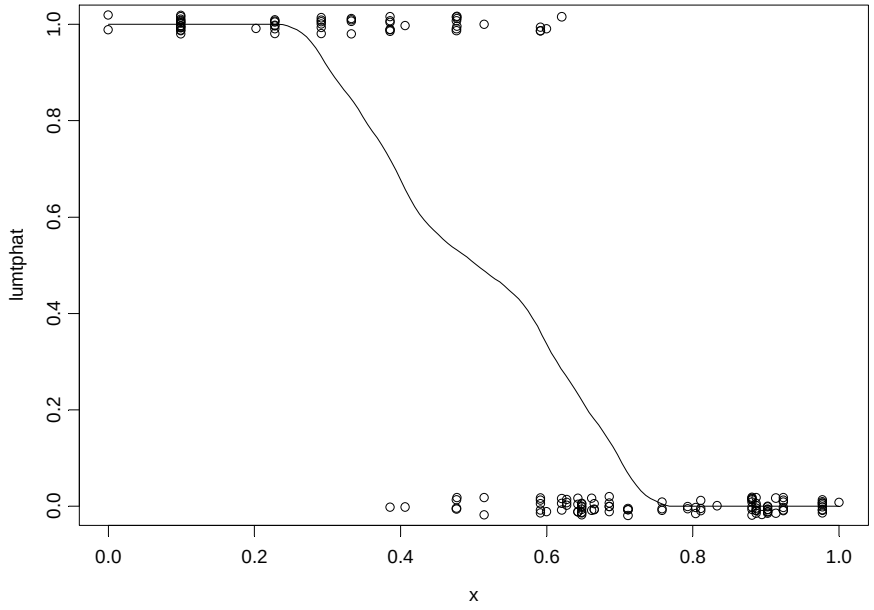

Figure 28. Gasser-Müller Estimator with Epanechnikov Kernel (i.e., $K(u) = .75(1-u^2)$, for $-1 \leq u \leq 1$) and a Bandwidth of 0.15.

The simulations presented in Chapter 4 have indicated that the bias correction methods of Hart and M & S create more accurate estimates and confidence intervals than the biased M & S method. To compute the 95% confidence intervals for LD_{20}, LD_{80}, and the cline width, we used the Hart *Bcn* method and the M & S (bias-corrected) method. The 95% *Bcn* confidence intervals for LD_{20}, LD_{80}, and the cline width are computed as

$$\left(\hat{\theta}_\alpha + \frac{B_{C,n,\theta_\alpha} \sqrt{Var[\hat{p}(\theta_\alpha)]}}{p'(\theta_\alpha)} \right) \pm 1.96 \sqrt{\frac{Var[\hat{p}(\theta_\alpha)]}{[p'(\theta_\alpha)]^2}}$$

and

$$\left(\hat{\theta}_{.8}+\frac{B_{C,n,\theta_8}\sqrt{Var[\hat{p}(\theta_{.8})]}}{p'(\theta_{.8})}\right)-\left(\hat{\theta}_{.2}+\frac{B_{C,n,\theta_2}\sqrt{Var[\hat{p}(\theta_{.2})]}}{p'(\theta_{.2})}\right)$$
$$\pm 1.96\sqrt{\frac{Var[\hat{p}(\theta_{.2})]}{[p'(\theta_{.2})]^2}+\frac{Var[\hat{p}(\theta_{.8})]}{[p'(\theta_{.8})]^2}-2Cov(\theta_{.2},\theta_{.8})},$$

where α = 0.2 for LD_{20} and 0.8 for LD_{80}; $B_{C,n}$ is computed as

$$B_{C,n,\theta_\alpha}=\frac{C^3 n^{-3/5}\sigma_K^2}{2\left(\sum_{i=1}^{n}\left\{\int_{s_{i-1}}^{s_i}K\left[n^{1/5}(\theta_\alpha-u)/C\right]du\right\}^2\right)^{1/2}}\frac{p''(\theta_\alpha)}{\sigma}.$$

The 95% M & S (bias-corrected) confidence intervals for LD_{20}, LD_{80}, and the cline width are computed as

$$\left(\hat{\theta}_{\alpha}+\frac{h^2 p''(\theta_\alpha)\sigma_K^2}{2p'(\theta_\alpha)}\right)\pm 1.96\sqrt{\frac{\alpha(1-\alpha)J_K}{nhp'(\theta_\alpha)^2}}$$

and

$$\left(\hat{\theta}_{.8}+\frac{h^2 p''(\theta_{.8})\sigma_K^2}{2p'(\theta_{.8})}\right)-\left(\hat{\theta}_{.2}+\frac{h^2 p''(\theta_{.2})\sigma_K^2}{2p'(\theta_{.2})}\right)$$
$$\pm 1.96\sqrt{\frac{\alpha(1-\alpha)J_K}{nh}\left(\frac{1}{p'(\theta_{.2})^2}+\frac{1}{p'(\theta_{.8})^2}\right)-2Cov(\theta_{.2},\theta_{.8})}.$$

In the Hart *Bcn* method and in the M & S (bias-corrected) confidence interval forumula, we need to compute the first derivative and the second derivative of *x* at θ_α.

To estimate $p'(x)$ and $p''(x)$, we used the first and second difference method as follows:

$$\hat{p}'(x)=\frac{\hat{p}(x_i)-\hat{p}(x_{i-1})}{\Delta x},$$

$$\hat{p}''(x)=\frac{\hat{p}'(x_i)-\hat{p}'(x_{i-1})}{\Delta x}.$$

We evaluated the goodness of the second difference method using a known probit distribution with parameters μ = .5 and σ = .1. Further, we evaluated the probit curve at a sequence of equally spaced points. The second derivative of the probit distribution at LD_{20} and LD_{80} are 23.562 and -23.562, respectively. We compared these second derivatives with those computed using the second difference method for various sample sizes. The results are shown in Table 7.

Table 7. Estimated Second Derivative of *p(x)* for the Probit

n	2nd derivative at LD_{20}	2nd derivative at LD_{80}
12	12.535	-12.331
24	15.928	-16.099
48	16.279	-17.520
96	18.334	-20.884
120	19.568	-18.993
400	20.328	-19.900

As the sample size increases, the estimated $p''(\theta_\alpha)$ becomes closer to the true $p''(\theta_\alpha)$ of the probit distribution. For a sample size of 400, we also compared the estimated B_{C,n,θ_α} with the true B_{C,n,θ_α}. As shown in Table 8, the estimated B_{C,n,θ_α} are very close to the true B_{C,n,θ_α}.

The second difference method requires the computation of $\hat{p}(x)$. However, the computation of $\hat{p}(x)$ requires the knowledge of the bandwidth *h*, and the optimal *h* requires the knowledge of $p''(x)$. The problem we are facing is that we have two

unknowns here (h and $p''(x)$), both need to be estimated and they depend on each other. Our approach is to use some reasonable bandwidth to compute $\hat{p}(x)$ first. With $\hat{p}(x)$, we then estimated $p''(x)$. Using $p''(x)$, we estimated the optimal h. Finally, using the optimal h, we recalculated $\hat{p}(x)$, $p'(x)$, and $p''(x)$.

Table 8. Comparison of Estimated and True B_{C,n,θ_α}

	LD_{20}		LD_{80}	
Method	$p''(\theta_{.2})$	$B_{C,n,\theta_{.2}}$	$p''(\theta_{.8})$	$B_{C,n,\theta_{.8}}$
true	23.562	0.4981	-23.562	-0.4981
2nd difference	20.328	0.4981	-19.900	-0.4876

We used an optimal bandwidth of 0.108 (initial bandwidth = 0.12) to estimate the LD_{20}, LD_{80}, and the cline width for the *lumt* distances. As stated earlier, the LD_{20}, LD_{80}, and the cline width for the *lumt* data are themselves estimates subject to bias as proven by the theory we presented. Both the *Bcn* and the M & S biased corrected estimates are attempts at correcting the bias of the estimates. The 95% confidence intervals for LD_{20}, LD_{80}, and the cline width for the *lumt* sample using the *Bcn* method and the M & S (bias-corrected) method are shown in Table 9.

Both the Hart *Bcn* method and the M & S (bias-corrected) method adjust the negative bias of the *lumt* LD_{20} estimate and produce larger LD_{20} estimates than the *lumt* LD_{20} estimate. On the other hand, both methods correct the positive bias of the *lumt* LD_{80} estimate and the resulting bias-corrected LD_{80} estimates are smaller than the *lumt* LD_{80}

estimate. The cline width has negative bias and both bias-corrected methods attempt to correct the bias by increasing the cline width estimate.

Table 9. 95% Confidence Intervals for LD_{20}, LD_{80}, Cline Width for *lumt* Distance

		Bcn		M & S (bias-corrected)	
	lumt	Estimate	95% confidence interval	Estimate	95% confidence interval
LD_{20}	0.6528	0.6569	(0.6381, 0.6757)	0.6689	(0.6342, 0.7036)
LD_{80}	0.3532	0.3455	(0.3233, 0.3676)	0.3358	(0.2948, 0.3767)
Cline width	0.2995	0.3114	(0.2823, 0.3405)	0.3331	(0.2795, 0.3868)

The Hart *Bcn* method produces shorter interval length than the M & S (bias-corrected) method. In fact, the *Bcn* confidence intervals lie completely within the M & S (bias-corrected) intervals.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

In our research, we have developed the distributional properties of LD_{20}, LD_{80}, and the cline width. The bias of the estimators for LD_{20}, LD_{80}, and the cline width depends on the bandwidth, the sample size, the curvature of the cline $p''(x)$, and the variance of the kernel. When the bandwidth is given, the bias does not depend on the design density. However, if a data set is given and an optimal bandwidth is computed, the bias depends on the design density. When $p''(x)$ is positive, the bias is negative and the estimator tends to underestimate the LD. Conversely, the bias is positive and the estimator overestimates the LD when $p''(x)$ is negative. For the bias to be zero, it is necessary for the bandwidth to tend to zero. For the variance to tend to zero, we need the sample size to tend to ∞. For both the bias and the variance to tend to zero, we need nh to tend to ∞. Further, the variance is inversely proportional to the design density and the sample size.

Based on the distributional properties of LD_{20}, LD_{80}, and the cline width, we developed approaches for constructing confidence intervals for LD_{20}, LD_{80}, and the cline width using the Müller and Schmitt method (1988) and the Hart method (1997). In the simulation study mimicking a real data example (Jaarola, 1997), we generated a data set containing 156 observations from a probit distribution with equally spaced design points. Using 1,000 replicate samples generated from the parent probit data, we verified the

distributional properties of LD_{20}, LD_{80}, and the cline width. The simulated mean and variance of the estimates conform to the theoretical mean and variance.

We evaluated the performance of the confidence intervals based on the interval midpoint, the interval length, and the coverage probability. For various sample sizes, design points, and kernels, the Hart method and the Müller and Schmitt (bias-corrected) method outperforms the Müller and Schmitt method (without bias correction). The *Bnh* and the *Bcn* of the Hart method produce similar estimates and coverage probability. The mean interval lengths for the two methods are identical. The sample size, the bandwidth, and the design points affect the quality of the confidence intervals. A larger sample size (which results in a smaller bandwidth) improves the performance of the confidence intervals. When the distances are uniformly spaced, the confidence intervals are more superior.

Since the Hart method and the Müller and Schmitt (bias-corrected) method produce better confidence intervals than the biased Müller and Schmitt method, we used the Hart *Bcn* method and the M & S (bias-corrected) method to compute the confidence intervals for LD_{20}, LD_{80}, and the cline width of the mitochondrial DNA genetic data for the field voles. Both methods are comparable in correcting the bias of the estimates.

Our simulation indicates that the covariance of LD_{20} and LD_{80} is quite small. One of the areas for future research is to explicitly express the covariance in terms of the sample size, the spatial sampling design density, the bandwidth, the variance, and the kernel, and to observe how the covariance varies with these characteristics. Based on our knowledge of the variance of the kernel estimator (Hart, 1997)

$$Var(\hat{p}(x)) = \frac{\sigma^2}{nh}\frac{1}{f(x)}\int_{-1}^{1} K^2(u)\,du + O\left(\frac{1}{n}\right) + O\left(\frac{1}{n^2h^2}\right),$$

we conjecture that the covariance of $\hat{p}(LD_{20}, LD_{80})$ is positively related to the variance, and inversely proportional to the sample size, bandwidth, and the design density.

This dissertation focuses on a response curve which increases or decreases monotonically. We have demonstrated that the kernel estimation performs well with a monotonic curve. For sufficiently large sample size (156 in our field vole genetic data) the kernel estimate is monotone. However, the kernel estimate is not necessarily monotone for finite samples, and $LD_{100\alpha}$ might not be defined. To solve the non-monotonicity problem, methods for defining monotonized kernel estimates are needed. One such method (Müller and Schmitt, 1988) is to find for a given *y* coordinate ($0 \le \alpha \le 1$) the corresponding *x* coordinate of the graph of the function estimate, take the average of the smallest and the largest of all *x* coordinates, where the kernel estimate is equal to α and the estimate of the first derivative

$$\hat{p}^{(1)}(x) = \frac{1}{b^2}\sum_{i=1}^{n}\int_{s_{i-1}}^{s_i} K^{(1)}\left(\frac{x-u}{b}\right)duy_i$$

is positive. The graph is defined by

$$\tilde{p} \equiv \left\{(\theta_\alpha, \alpha) \mid \alpha \in [0,1], \hat{\theta}_\alpha = \frac{1}{2}\left(\inf M_\alpha + \sup M_\alpha\right)\right\},$$

where $M_\alpha = \{x : \hat{p}(x) = \alpha,\ \hat{p}^{(1)}(x) > 0\}$.

Further work is needed to investigate the properties of cline width using this procedure.

The *Bcn* method and the M & S (bias-corrected) method require the knowledge of $p'(x)$ and $p''(x)$ in the calculation of confidence intervals. In this dissertation, we have

used the second difference method for estimating the derivatives. Further investigation is needed to evaluate other methods such as the higher order kernel method as described in Hart (1997).

In genetics fieldwork, samples are mostly collected at fixed locations. The distance between a starting point and these fixed points follows a discrete distribution. We used the kernel density estimation for smoothing the probability density function of the distance. Further research and techniques are needed to estimate LD_{20}, LD_{80}, and the cline width for discrete location sampling. It might also be of interest to match the experimental design with data analysis and to develop an optimal design to obtain the best estimates of LD_{20}, LD_{80}, and the cline width.

In conclusion, our research has produced a new tool for estimating the cline width and its confidence interval using kernel techniques. This cline width provides valuable information to biologists regarding the extent of a zone where two species might hybridize, yielding crucial data for genetic diversity.

APPENDIX A

S-PLUS PROGRAMS FOR SIMULATION STUDY

This Appendix shows the codes for creating the simulation results presented in Chapter 4. The purposes of the simulation are twofold: (1) to investigate the distributional results of LD_{20}, LD_{80}, and the cline width for the probit curve; (2) to evaluate the performance of the confidence intervals for LD_{20}, LD_{80}, and the cline width.

Creating Probit Data Set

The following codes create a probit data set with parameters mean, $\mu = 0.5$ and standard deviation, $\sigma = 0.1$ that contains 156 observations and to compute the optimal bandwidth. The distance is equally spaced.

```
n <- 156
m <- 1
m2 <- 1000
sigma <- 0.1
mu <- 0.5
alpha20 <- 0.2
alpha80 <- 0.8

LD20 <- qnorm(alpha20,mu,sigma)
pLD20 <- pnorm((LD20 - mu)/sigma)
p1LD20 <- dnorm(LD20,mu,sigma)
p2LD20 <-  -(1/sqrt(2 * pi)) * (1/sigma^2) * exp(-0.5 * ((LD20 -
    mu)/sigma)^2) * ((LD20 - mu)/sigma)

LD80 <- qnorm(alpha80,mu,sigma)
pLD80 <- pnorm((LD80 - mu)/sigma)
p1LD80 <- dnorm(LD80,mu,sigma)
```

```
p2LD80 <-  -(1/sqrt(2 * pi)) * (1/sigma^2) * exp(-0.5 * ((LD80 -
    mu)/sigma)^2) * ((LD80 - mu)/sigma)

# Kernel K21: 3/4*(1-x^2)
B <- integrate(function(x){((3/4) * (1 - x^2) * x^2)} /2, -1, 1)$integral
V <- integrate(function(x){((3/4)*(1-x**2))**2},-1,1)$integral

C1 <- ((alpha20*(1-alpha20)*V)/(4*p2LD20^2*B^2))^.2

# b is the bandwidth
b <- C1/n^.2

i <- 1:n
xi <- (i-1)/(n-1)
quantile <- (xi-mu)/sigma
probit <- pnorm(quantile)
wi <- seq(0,156,length=500)/156
nw <-  length(wi)
den <- density(xi,n=nw,from=0,to=1)
x <- den$x
f1x <- den$y
mx <- max(diff(x))
gx <- sum(f1x)*mx
fx <-- f1x/gx
yi <- rbinom(1*n,1,probit)
```

Gasser and Müller Kernel Estimator

The following codes compute the Gasser and Müller kernel estimator,

$\hat{p}(x) = \frac{1}{h}\sum_{i=1}^{n} y_i \int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right) du$ for the probit data, and estimate LD20, LD80, $p'(x)$,

$p''(x)$. The program also computes

$$Var[\hat{p}(\theta_\alpha)] = \hat{p}(\theta_\alpha)[1-\hat{p}(\theta_\alpha)]\frac{1}{h^2}\sum_{i=1}^{n}\left(\int_{s_{i-1}}^{s_i} K\left(\frac{\theta_\alpha - u}{h}\right) du\right)^2 \text{ and}$$

$$Var(\hat{p}(x)) = \frac{\sigma^2}{nh}\frac{1}{f(x)}J_K + O\left(\frac{1}{n}\right).$$

```
    xi1 <- xi[2:n]
    xi1 <- c(xi1,1)

    si <- (xi+xi1)/2
    si0 <- c(0,si[1:n-1])

    xk <- -t(outer(x,c(0,si),"-")/b)
    k <- abs(xk)<=1
    int <- t(matrix(0,nw,n+1))
    wint <- matrix(0,n,nw)

for (j in 1:nw){
      xk <- ifelse(xk<=(-1),-1,xk)
      xk <- ifelse(xk>=1,1,xk)
      int[,j] <- 3/4*((xk[,j]-(xk[,j]^3)/3)-((-1)-((-1)^3)/3))
      wint[,j] <- diff(int[,j])
           }

    ld20 <- rep(0,m)
    ld80 <- rep(0,m)

    phat <- matrix(0,nw,m)
    p1hatx <- matrix(0,nw-1,m)
    p2hatx <- matrix(0,nw-2,m)

    pld20 <- rep(0,m)
    pld80 <- rep(0,m)

    p1hat20 <- rep(0,m)
    p2hat20 <- rep(0,m)

    p1hat80 <- rep(0,m)
    p2hat80 <- rep(0,m)

    var20 <- rep(0,m)
    var80 <- rep(0,m)

    for(k in 1:m) {
```

```
        wint <- sweep(wint,2,apply(wint,2,sum),"/")

        phat[,k] <- as.vector(yi%*%wint)

        ld20[k] <- approx(phat[,k],x,alpha20)$y
        p1hatx[,k] <- diff(phat[,k])/max(diff(x))
        p1hat20[k] <- approx(x[-1],p1hatx[,k],ld20[k])$y
        p2hatx[,k] <- diff(p1hatx[,k])/max(diff(x))
        p2hat20[k] <- approx(x[c(1,length(x))],p2hatx[,k],ld20[k])$y
        pld20[k] <- approx(x,phat[,k],LD20)$y

        ld80[k] <- approx(phat[,k],x,alpha80)$y
        p1hat80[k] <- approx(x[-1],p1hatx[,k],ld80[k])$y
        p2hat80[k] <- approx(x[c(1,length(x))],p2hatx[,k],ld80[k])$y
        pld80[k] <- approx(x,phat[,k],LD80)$y
    }

for(k in 1:m) {

        xk120 <-  -(LD20-c(0, si))/b
        xk120 <- ifelse(xk120 <= (-1), -1, xk120)
        xk120 <- ifelse(xk120 >= 1, 1, xk120)
        int120 <- (3/4) * ((xk120 - (xk120^3)/3) - ((-1) -  ((-1)^3)/3))
        wint120 <- diff(int120)
        wint1220 <- sum(wint120^2)
        var20[k] <- pld20[k]*(1-pld20[k])*wint1220

        xk180 <-  -(LD80-c(0, si))/b
        xk180 <- ifelse(xk180 <= (-1), -1, xk180)
        xk180 <- ifelse(xk180 >= 1, 1, xk180)
        int180 <- (3/4) * ((xk180 - (xk180^3)/3) - ((-1) -  ((-1)^3)/3))
        wint180 <- diff(int180)
        wint1280 <- sum(wint180^2)
        var80[k] <- pld80[k]*(1-pld80[k])*wint1280

}

# for computing O(1/n) in var(p(x))

d20 <- 1 #density of uniform = 1
d80 <- 1
```

```
w1 <- seq(-1,1,.01)
den20 <- density(ld20-b*w1,n=length(w1))

fO20 <- ((3/4)*(1-w1^2))^2/(den20$y)
int120 <- sum(fO20[2:(length(fO20)-1)])*max(diff(w1))
int220 <- V/d20

den80 <- density(lumtld80-b*w1,n=length(w1))
fO80 <- ((3/4)*(1-w1^2))^2/(den80$y)
int180 <- sum(fO80[2:(length(fO80)-1)])*max(diff(w1))
int280 <- V/d80

f20 <- rep(0,m)
O1overnvar20 <- rep(0,m)
v20 <- rep(0,m)
f80 <- rep(0,m)
O1overnvar80 <- rep(0,m)
v80 <- rep(0,m)

for(k in 1:m) {

      f20[k] <- approx(x,phat[,k],ld20[k])$y
      O1overnvar20[k] <- (f20[k]*(1-f20[k])/(n*b))*(int120-int220)
      v20[k] <- f20[k]*(1-f20[k])*V/(n*b*d20)+O1overnvar20[k]

      f80[k] <- approx(x,phat[,k],ld80[k])$y
      O1overnvar80[k] <- (f80[k]*(1-f80[k])/(n*b))*(int180-int280)
      v80[k] <- f80[k]*(1 - f80[k])*V/(n*b*d80)+    O1overnvar80[k]
}
```

Computing *Bcn*

The following codes compute $B_{C,n} = \dfrac{C^3 n^{-3/5}\sigma_K^2 p''(x)}{2\sigma\left(\sum_{i=1}^{b}\left\{\int_{s_{i-1}}^{s_i} K\left[n^{1/5}(x-u)/C\right]du\right\}^2\right)^{1/2}}$ for the probit data.

```
xi1 <- xi[2:n]
xi1 <- c(xi1,1)
si <- (xi+xi1)/2
```

```
si0 <- c(0,si[1:n-1])

sig2k <- 2*B
```

compute *Bcn* for LD20

```
C1 <- ((alpha20*(1-alpha20)*V)/(4*p2LD20^2*B^2))^.2
b <- C1/n^.2
xk120 <- -(LD20-c(0, si))/b
xk120 <- ifelse(xk120 <= (-1), -1, xk120)
xk120 <- ifelse(xk120 >= 1, 1, xk120)
int120 <- (3/4) * ((xk120 - (xk120^3)/3) - ((-1)- ((-1)^3)/3))
wint120 <- diff(int120)
wint1220 <- sum((b*wint120)^2)
v320  <-  pLD20*(1-pLD20)*wint1220
Bcn220 <- (C1^3 * n^(-0.6) * 2 * B * p2LD20)/(2 *
   sqrt(wint1220 * alpha20 * (1-alpha20)))
```

compute *Bcn* for LD80

```
xk180 <- -(LD80-c(0, si))/b
xk180 <- ifelse(xk180 <= (-1), -1, xk180)
xk180 <- ifelse(xk180 >= 1, 1, xk180)
int180 <- (3/4) * ((xk180 - (xk180^3)/3) - ((-1) -  ((-1)^3)/3))
wint180 <- diff(int180)
wint1280 <- sum((b*wint180)^2)
v380  <-  pLD80*(1-pLD80)*wint1280
Bcn280[ik] <- (C1^3 * n^(-0.6) * 2 * B * p2LD80)/(2 *
   sqrt(wint1280 * alpha80* (1-alpha80)))
```

Bias of the Gasser- Müller Kernel Estimator

The following codes compute the bias of the Gasser- Müller kernel estimator,

$$E(\hat{p}_h(x)) - p(x) = \frac{h^2}{2} p''(x)\sigma_K^2 + o(h^2) + O(n^{-1}).$$

compute $o(h^2), O(n^{-1})$ for LD20

```
x <- seq(0,1,0.01)

f20 <- function(u){ifelse(abs((LD20-u)/b)<=1,3/4*(1-abs((LD20-
   u)/b)^2),0)}
```

```
ff20 <- function(u){ifelse(abs((LD20-u)/b)<=1,
   3/4*(1-abs((LD20-u)/b)^2),0) * pnorm((u-mu)/sigma)}

ag20 <- rep(0,n)
ah20 <- rep(0,n)
for (i in 1:n){
   ag20[i] <- pxi[i]*integrate(f20,lower=si0[i],upper=si[i],
   LD20=LD20,b=b)$integral
   ah20[i] <- integrate(ff20,lower=si0[i],upper=si[i],
   LD20=LD20,b=b,mu=mu,sigma=sigma)$integral
}
O1overn20 <- sum(ag20-ah20)/b

o1int20 <- integrate(function(v){v^2 * exp(-0.5 * ((LD20-
mu)/sigma)^2)*(-exp(-0.5*((-2*(LD20-mu)*b*v+b^2*v^2)/sigma))*
((LD20-mu-b*v)/sigma)+ ((LD20-mu)/sigma))*((3/4)*(1-v**2))},
-1,1)$integral

ob220 <- (1/sqrt(2 * pi)) * (1/sigma^2) *o1int20*b^2/2


# compute o(h^2),O(n^-1) for LD80
f80 <- function(u){ifelse(abs((LD80-u)/b)<=1,3/4*
   (1-abs((LD80-u)/b)^2),0)}

ff80 <- function(u){ifelse(abs((LD80-u)/b)<=1,3/4*
   (1-abs((LD80-u)/b)^2),0) * pnorm((u-mu)/sigma)}

ag80 <- rep(0,n)
ah80 <- rep(0,n)
for (i in 1:n){
   ag80[i] <- pxi[i]*integrate(f80,lower=si0[i],upper=si[i],
   LD80=LD80,b=b)$integral
   ah80[i] <- integrate(ff80,lower=si0[i],upper=si[i],
   LD80=LD80,b=b,mu=mu,sigma=sigma)$integral
}
O1overn80 <- sum(ag80-ah80)/b

o1int80 <- integrate(function(v){v^2 * exp(-0.5 *  ((LD80 -
mu)/sigma)^2)*(-exp(-0.5*((-2*(LD80-mu)*b*v+b^2*v^2)/sigma))*
((LD80-mu-b*v)/sigma)+((LD80-mu)/sigma))*((3/4)*(1-v**2))},
-1,1)$integral
```

```
ob280 <- (1/sqrt(2 * pi)) * (1/sigma^2) *o1int80*b^2/2

sig2k <- 2*B
bias20 <- b^2*p2LD20*sig2k/2+O1overn20+ob220
bias80 <- b^2*p2LD80*sig2k/2+O1overn80+ob280
```

Covariance of LD20 and LD80 for the Kernel Estimator

The following codes compute Cov($\theta_{.2}, \theta_{.8}$) and cov($\hat{p}(\theta_{.2}), \hat{p}(\theta_{.8})$).

```
# compute expected value and variance of  p̂(LD20)

a20 <- rep(0,n)
ag20 <- rep(0,n)
for (i in 1:n){
   a20[i] <-
   integrate(f20,lower=si0[i],upper=si[i],LD20=LD20,b=b)$integral
   ag20[i] <- pxi[i]*a20[i]
}
EpLD20 <- sum(ag20)/b
VpLD20 <- alpha20*(1-alpha20)*sum(a20^2)/(b^2)

# compute expected value and variance of  p̂(LD80)

a80 <- rep(0,n)
ag80 <- rep(0,n)
for (i in 1:n){
   a80[i] <-
   integrate(f80,lower=si0[i],upper=si[i],LD80=LD80,b=b)$integral
   ag80[i] <- pxi[i]*a80[i]
}
EpLD80 <- sum(ag80)/b
VpLD80 <- alpha80*(1-alpha80)*sum(a80^2)/(b^2)

# compute expected value of p(LD20) and p(LD80)
eg2080 <- rep(0,n)
for (i in 1:n){
   eg2080[i] <- (pxi[i]*(1-pxi[i])+(pxi[i]^2))*(integrate(f20,lower=si0[i],
   upper=si[i],LD20=LD20,b=b)$integral)*
      (integrate(f80,lower=si0[i],upper=si[i],
```

```
    LD80=LD80,b=b)$integral)
}

cg2080 <- matrix(0,n,n)
for (i in 1:n){
    for (j in 1:n){
    cg2080[i,j] <-
    ifelse(i==j,0,(pxi[i]*pxi[j]*(integrate(f20,lower=si0[i],upper=si[i],
    LD20=LD20,b=b)$integral)*(integrate(f80,lower=si0[j],
    upper=si[j],LD80=LD80,b=b)$integral)))
}
    }

Ep20p80 <- (1/b^2)*(sum(eg2080) + sum(cg2080))
Covp20p80 <- Ep20p80 - EpLD20 * EpLD80

# compute cov(LD20,LD80)
covLD20LD80 <- Covp20p80/(p1LD20*p1LD80)

# compute var(LD20) and var(LD80)
varLD20 <- VpLD20/(p1LD20^2)
varLD80 <- VpLD80/(p1LD80^2)

# compute correlation of LD20 and LD80
corLD20LD80 <- covLD20LD80/(sqrt(varLD20)*sqrt(varLD80))
```

Replicate Samples

The following codes generate 1,000 replicate samples and compute the kernel estimators.

```
m2 <- 1000
b1 <- 0.123

phat <-  matrix(0,nw,m2)

xk <-  -t(outer(x,c(0,si),"-")/b1)
int <- t(matrix(0,nw,n+1))
    wint <- matrix(0,n,nw)

    for (j in 1:nw){
      xk <- ifelse(xk<=(-1),-1,xk)
```

```
        xk <- ifelse(xk>=1,1,xk)
        int[,j] <- 3/4*((xk[,j]-(xk[,j]^3)/3)-((-1)-((-1)^3)/3))
        wint[,j] <- diff(int[,j])
                }

# generate replicate samples
for(k in 1:m2) {
    yi <- NULL
    phat <- matrix(0,nw,m2)

    yi <- c(yi,rbinom(1*n,1,probit))
    wint <- sweep(wint,2,apply(wint,2,sum),"/")
    phat[,k] <- as.vector(yi%*%wint)
        }
```

Confidence Intervals

The following codes compute 95% confidence intervals using Müller and Schmitt methods and Hart methods and compute coverage probability.

```
n <- 100
m <- 4000
sigma <- 0.1
mu <- 0.5
alpha20 <- 0.2
alpha80 <- 0.8

LD20 <- qnorm(alpha20)*sigma+mu
pLD20 <- pnorm((LD20 - mu)/sigma)
p1LD20 <- dnorm(LD20,mu,sigma)
p2LD20 <-  -(1/sqrt(2 * pi)) * (1/sigma^2) *
exp(-0.5 * ((LD20 - mu)/sigma)^2) * ((LD20 - mu)/sigma)

LD80 <- qnorm(alpha80)*sigma+mu
pLD80 <- pnorm((LD80 - mu)/sigma)
p1LD80 <-- dnorm(LD80,mu,sigma)
p2LD80 <-  -(1/sqrt(2 * pi)) * (1/sigma^2) *
exp(-0.5 * ((LD80 - mu)/sigma)^2) * ((LD80 - mu)/sigma)

# Kernel K21: 3/4*(1-x^2)
```

```
B <- integrate(function(x){((3/4) * (1 - x^2) * x^2)}/2, -1, 1)$integral
V <- integrate(function(x){((3/4)*(1-x**2))**2},-1,1)$integral

beta <- 0.05

C1 <- ((alpha20*(1-alpha20)*V)/(4*p2LD20^2*B^2))^.2

b <- C1/n^.2

    f20 <- rep(0,m)
    v20 <- rep(0,m)
    LD20 <- rep(0,m)
    LD20plus <- rep(0,m)
    LD20minus <- rep(0,m)
    LD20plusH <- rep(0,m)
    LD20minusH <- rep(0,m)

    f80 <- rep(0,m)
    v80 <- rep(0,m)
    LD80 <- rep(0,m)

    LD80plus <- rep(0,m)
    LD80minus <- rep(0,m)
    LD80plusH <- rep(0,m)
    LD80minusH <- rep(0,m)

    i <- 1:n
    xi <- (i-1)/(n-1)
    xi1 <- xi[2:n]
    xi1 <- c(xi1,1)

    si <- (xi+xi1)/2
    si0 <- c(0,si[1:n-1])

    quantile <- (xi-mu)/sigma
    probit <- pnorm(quantile)

    x <- seq(0,1,0.01)

    px <- pnorm((x-mu)/sigma)

    nx <- length(xi)
```

```
    phat <- matrix(0,nx,m)
    xk <-  -t(outer(xi,c(0,si),"-")/b)
    k <- abs(xk)<=1
    int <- t(matrix(0,nx,n+1))
    wint <- matrix(0,n,nx)


# compute kernel estimator for LD20 and LD80

    for (j in 1:nx){
      xk <- ifelse(xk<=(-1),-1,xk)
      xk <- ifelse(xk>=1,1,xk)
      int[,j] <- (3/4)*((xk[,j]-(xk[,j]^3)/3)-((-1)-((-1)^3)/3))
      wint[,j] <- diff(int[,j])
       }

    phat <- matrix(0,nx,m)
    yi <- matrix(0,n,m)
    p1hatx <- matrix(0,nx-1,m)
    p2hatx <- matrix(0,nx-2,m)
    p1hat20 <- rep(0,m)
    p2hat20 <- rep(0,m)

    p1hat80 <- rep(0,m)
    p2hat80 <- rep(0,m)

    for(k in 1:m) {
        yi[,k] <- rbinom(1*n,1,probit)
        wint <- sweep(wint,2,apply(wint,2,sum),"/")
        phat[,k] <- as.vector(yi[,k]%*%wint)
        LD20[k] <- approx(phat[,k],xi,alpha20)$y
        p1hatx[,k] <- diff(phat[,k])/max(diff(xi))
        LD80[k] <- approx(phat[,k],xi,alpha80)$y
}
for(k in 1:m) {
    p1hat20[k] <- approx(xi[-1],p1hatx[,k],LD20[k])$y
    p2hatx[,k] <- diff(p1hatx[,k])/max(diff(xi))
    p2hat20[k] <- approx(xi[-c(1,length(xi))], p2hatx[,k],LD20[k])$y

    f20[k] <- approx(xi,phat[,k],qnorm(alpha20,.5,sigma))$y
    v20[k] <- f20[k]*(1-f20[k])*sum(wint[,kk]^2)
```

```
    v120 <- alpha20*(1-alpha20)*sum(wint[,kk]^2)
    LD20plus[k] <- approx(phat[,k],xi,alpha20+sqrt(v120))$y
    LD20minus[k] <- approx(phat[,k],xi,alpha20-sqrt(v120))$y
    LD20plusH[k] <- approx(phat[,k],xi,alpha20+sqrt(v20[k]))$y
    LD20minusH[k] <- approx(phat[,k],xi,alpha20-sqrt(v20[k]))$y

    p1hat80[k] <- approx(xi[-1],p1hatx[,k],LD80[k])$y
    p2hat80[k] <- approx(xi[-c(1,length(xi))], p2hatx[,k],LD80[k])$y

    f80[k] <- approx(xi,phat[,k],qnorm(alpha80,.5,sigma))$y
    v80[k] <- f80[k]*(1-f80[k])*sum(wint[,kk]^2)
    v180 <- alpha80*(1-alpha80)*sum(wint[,kk]^2)
    LD80plus[k] <- approx(phat[,k],xi,alpha80+sqrt(v180))$y
    LD80minus[k] <- approx(phat[,k],xi,alpha80-sqrt(v180))$y
    LD80plusH[k] <- approx(phat[,k],xi,alpha80+sqrt(v80[k]))$y
    LD80minusH[k] <- approx(phat[,k],xi,alpha80-sqrt(v80[k]))$y
    }

HLD20 <- rep(0,m)
HloLD20 <- rep(0,m)
HupLD20 <- rep(0,m)
Hbcn2LD20 <- rep(0,m)
Hbcn2loLD20 <- rep(0,m)
Hbcn2upLD20 <- rep(0,m)

HbnhLD20 <- rep(0,m)
HbnhloLD20 <- rep(0,m)
HbnhupLD20 <- rep(0,m)
HbnhaLD20 <- rep(0,m)
HbnhaloLD20 <- rep(0,m)
HbnhaupLD20 <- rep(0,m)

v220 <- rep(0,m)
v280 <- rep(0,m)

loLD20 <- rep(0,m)
upLD20 <- rep(0,m)
bcLD20 <- rep(0,m)
lobcLD20 <- rep(0,m)
upbcLD20 <- rep(0,m)
DQloLD20 <- rep(0,m)
```

```
DQupLD20 <- rep(0,m)

HLD80 <- rep(0,m)
HloLD80 <- rep(0,m)
HupLD80 <- rep(0,m)
Hbcn2LD80 <- rep(0,m)
Hbcn2loLD80 <- rep(0,m)
Hbcn2upLD80 <- rep(0,m)

HbnhLD80 <- rep(0,m)
HbnhloLD80 <- rep(0,m)
HbnhupLD80 <- rep(0,m)
HbnhaLD80 <- rep(0,m)
HbnhaloLD80 <- rep(0,m)
HbnhaupLD80 <- rep(0,m)

loLD80 <- rep(0,m)
upLD80 <- rep(0,m)
bcLD80 <- rep(0,m)
lobcLD80 <- rep(0,m)
upbcLD80 <- rep(0,m)
DQloLD80 <- rep(0,m)
DQupLD80 <- rep(0,m)

Hcline <- rep(0,m)
Hlocline <- rep(0,m)
Hupcline <- rep(0,m)

Hbcn2cline <- rep(0,m)
Hbcn2locline <- rep(0,m)
Hbcn2upcline <- rep(0,m)
Hbnhcline <- rep(0,m)
Hbnhlocline <- rep(0,m)
Hbnhupcline <- rep(0,m)

Hbnhacline <- rep(0,m)
Hbnhalocline <- rep(0,m)
Hbnhaupcline <- rep(0,m)

locline <- rep(0,m)
upcline <- rep(0,m)
```

```
bccline <- rep(0,m)
lobccline <- rep(0,m)
upbccline <- rep(0,m)

DQlocline <- rep(0,m)
DQupcline <- rep(0,m)

# Hart
HLD20 <- LD20 + hz220*sqrt(v20)/p1LD20
HloLD20 <- HLD20 - qnorm(1-beta/2)*(sqrt(v20)/p1LD20)
HupLD20 <- HLD20 + qnorm(1-beta/2)*(sqrt(v20)/p1LD20)
lenHLD20 <- HupLD20 - HloLD20
HLD80 <- LD80 + hz280*sqrt(v80)/p1LD80
HloLD80 <- HLD80 - qnorm(1-beta/2)*(sqrt(v80)/p1LD80)
HupLD80 <- HLD80 + qnorm(1-beta/2)*(sqrt(v80)/p1LD80)

Hcline <- HLD80 - HLD20
Hlocline <- Hcline - qnorm(1-beta/2)*
   sqrt(v20/p1LD20^2+v80/p1LD80^2-2*CovLD20LD80)
Hupcline <- Hcline + qnorm(1-beta/2)*
   sqrt(v20/p1LD20^2+v80/p1LD80^2-2*CovLD20LD80)

# Müller and Schmitt
v220 <- v120/(p1hat20^2)
loLD20 <- LD20 - qnorm(1-beta/2)*sqrt(v220)
upLD20 <- LD20 + qnorm(1-beta/2)*sqrt(v220)
v280 <- v180/(p1hat80^2)

loLD80 <- LD80 - qnorm(1-beta/2)*sqrt(v280)
upLD80 <- LD80 + qnorm(1-beta/2)*sqrt(v280)

v2cline <- v120/(p1hat20^2)+v180/(p1hat80^2)-2*CovLD20LD80
locline <- (LD80 - LD20)- qnorm(1-beta/2)*sqrt(v2cline)
upcline <- (LD80 - LD20)+ qnorm(1-beta/2)*sqrt(v2cline)

# Müller and Schmitt (Difference Quotient method)
DQloLD20 <- LD20 - qnorm(1-beta/2)*(LD20-LD20minus)
DQupLD20 <- LD20 + qnorm(1-beta/2)*(LD20plus-LD20)

DQloLD80 <- LD80 - qnorm(1-beta/2)*(LD80-LD80minus)
DQupLD80 <- LD80 + qnorm(1-beta/2)*(LD80plus-LD80)
```

```
DQlocline <- (LD80-LD20) - qnorm(1-beta/2)*sqrt((LD20-
    LD20minus)^2+(LD80-LD80minus)^2-2*CovLD20LD80)
DQupcline <- (LD80-LD20) + qnorm(1-beta/2)*sqrt((LD20plus-
    LD20)^2+(LD80plus-LD80)^2-2*CovLD20LD80)

# bias-corrected CI (Müller and Schmitt Theorem 4)
bcLD20 <- (LD20+(b^2*B*mu)/sigma^2)/(1+b^2*B/sigma^2)
bcsdLD20 <- (sqrt((alpha20*(1-alpha20)*V)/(n*b*p1LD20^2)))/
    (1+b^2*B/sigma^2)
lobcLD20 <- bcLD20 - (qnorm(1-beta/2))*bcsdLD20
upbcLD20 <- bcLD20 + (qnorm(1-beta/2))*bcsdLD20

bcLD80 <- (a$LD80+(b^2*B*mu)/sigma^2)/(1+b^2*B/sigma^2)
bcsdLD80 <- (sqrt((alpha80*(1-alpha80)*V)/(n*b*p1LD80^2)))/
    (1+b^2*B/sigma^2)
lobcLD80 <- bcLD80 - (qnorm(1-beta/2))*bcsdLD80
upbcLD80 <- bcLD80 + (qnorm(1-beta/2))*bcsdLD80

bccline <- bcLD80 - bcLD20
bcsdcline <- sqrt((alpha20*(1-alpha20)*V)/
    (n*b*p1LD20^2)/(1+b^2*B/sigma^2)^2 +(alpha80*
    (1-alpha80)*V)/(n*b*p1LD80^2)/(1+b^2*B/sigma^2)^2-
    2*CovLD20LD80)
lobccline <- bccline - (qnorm(1-beta/2))*bcsdcline
upbccline <- bccline + (qnorm(1-beta/2))*bcsdcline

# Hart (Bcn)
Hbcn2LD20 <- LD20 + Bcn220*sqrt(v20)/p1LD20
Hbcn2loLD20 <- Hbcn2LD20-qnorm(1-beta/2)*(sqrt(v20)/p1LD20)
Hbcn2upLD20 <- Hbcn2LD20+qnorm(1-beta/2)*(sqrt(v20)/p1LD20)

Hbcn2LD80 <- LD80 + Bcn280*sqrt(v80)/p1LD80
Hbcn2loLD80 <- Hbcn2LD80-qnorm(1-beta/2)*(sqrt(v80)/p1LD80)
Hbcn2upLD80 <- Hbcn2LD80+qnorm(1-beta/2)*(sqrt(v80)/p1LD80)

Hbcn2cline <- Hbcn2LD80 - Hbcn2LD20
Hbcn2locline <- Hbcn2cline - qnorm(1-beta/2)* sqrt(v20/p1LD20^2 +
    v80/p1LD80^2-2*CovLD20LD80)
Hbcn2upcline <- Hbcn2cline + qnorm(1-beta/2)* sqrt(v20/p1LD20^2 +
    v80/p1LD80^2-2*CovLD20LD80)

# Hart (Bnh) asymptotic
```

```
vxh20 <- alpha20*(1-alpha20)/(n*b)*V
Bnyasy20 <- (sum(ag20)/b-pLD20)/sqrt(vxh20)
HbnhaLD20 <- LD20 + Bnhasy20*sqrt(v20)/p1LD20
HbnhaloLD20 <- HbnhaLD20 - qnorm(1-beta/2)*(sqrt(v20)/p1LD20)
HbnhaupLD20 <- HbnhaLD20 + qnorm(1-beta/2)*(sqrt(v20)/p1LD20)

Vxh80 <- alpha80*(1-alpha80)/(n*b)*V
Bnyasy80 <- (sum(ag80)/b-pLD80)/sqrt(vxh80)
HbnhaLD80 <- LD80 + Bnhasy80*sqrt(v80)/p1LD80
HbnhaloLD80 <- HbnhaLD80 - qnorm(1-beta/2)*(sqrt(v80)/p1LD80)
HbnhaupLD80 <- HbnhaLD80 + qnorm(1-beta/2)*(sqrt(v80)/p1LD80)

Hbnhacline <- HbnhaLD80 - HbnhaLD20
Hbnhalocline <- Hbnhacline - qnorm(1-beta/2)* sqrt(v20/p1LD20^2 +
   v80/p1LD80^2 - 2*CovLD20LD80)
Hbnhaupcline <- Hbnhacline + qnorm(1-beta/2)* sqrt(v20/p1LD20^2 +
   v80/p1LD80^2 - 2*CovLD20LD80)

# Hart (Bnh)
Bnh20 <- (C1^3*n^(-.6)*p2LD20*2*B+O1overn20+ob220)/(2*sqrt(v320))
HbnhLD20 <- LD20 + Bnh20*sqrt(v20)/p1LD20
HbnhloLD20 <- HbnhLD20 - qnorm(1-beta/2)*(sqrt(v20)/p1LD20)
HbnhupLD20 <- HbnhLD20 + qnorm(1-beta/2)*(sqrt(v20)/p1LD20)

Bnh80 <- (C1^3*n^(-.6)*p2LD80*2*B+O1overn80+ob280)/(2*sqrt(v380))
HbnhLD80 <- LD80 + Bnh80*sqrt(v80)/p1LD80
HbnhloLD80 <- HbnhLD80 - qnorm(1-beta/2)*(sqrt(v80)/p1LD80)
HbnhupLD80 <- HbnhLD80 + qnorm(1-beta/2)*(sqrt(v80)/p1LD80)

Hbnhcline <-HbnhLD80 - HbnhLD20
Hbnhlocline <- Hbnhcline - qnorm(1-beta/2)* sqrt(v20/p1LD20^2 +
   a$v80/p1LD80^2 - 2*CovLD20LD80)
Hbnhupcline <- Hbnhcline + qnorm(1-beta/2)* sqrt(a$v20/p1LD20^2 +
   v80/p1LD80^2 - 2*CovLD20LD80)

qcline <- qnorm(alpha80,.5,sigma) - qnorm(alpha20,.5,sigma)

# LD20 coverage probability, M&S (5.4)
covLD20 <- sum(loLD20 < qnorm(alpha20,.5,sigma) &
   qnorm(alpha20,.5,sigma) < upLD20)/m

# LD20 coverage probability, M&S (5.7)
```

```
covDQLD20 <- sum(DQloLD20 < qnorm(alpha20,.5,sigma) &
   qnorm(alpha20,.5,sigma) < DQupLD20)/m

# LD20 coverage probability bias-correction, M&S Theorem 4"
covbcLD20 <- sum(lobcLD20 < qnorm(alpha20,.5,sigma) &
   qnorm(alpha20,.5,sigma) < upbcLD20)/m

# LD20 coverage probability, Hart (top of p.79)
covHLD20 <- sum(HloLD20 < qnorm(alpha20,.5,sigma) &
   qnorm(alpha20,.5,sigma) < HupLD20)/m

# LD20 coverage probability, Hart Bcn2
covHbcn2LD20 <- sum(Hbcn2loLD20 < qnorm(alpha20,.5,sigma) &
   qnorm(alpha20,.5,sigma) < Hbcn2upLD20)/m

# LD20 coverage probability, Hart Bnh
covHbnhLD20 <- sum(HbnhloLD20 < qnorm(alpha20,.5,sigma) &
   qnorm(alpha20,.5,sigma) < HbnhupLD20)/m

# LD20 coverage probability, Hart Bnh asymptotic
covHbnhaLD20 <- sum(HbnhaloLD20 < qnorm(alpha20,.5,sigma) &
   qnorm(alpha20,.5,sigma) < HbnhaupLD20)/m

# LD80 coverage probability M&S (5.4)
covLD80 <- sum(loLD80 < qnorm(alpha80,.5,sigma) &
   qnorm(alpha80,.5,sigma) < upLD80)/m

# LD80 coverage probability M&S (5.7)
covDQLD80 <- sum(DQloLD80 < qnorm(alpha80,.5,sigma) &
   qnorm(alpha80,.5,sigma) < DQupLD80)/m

# LD80 coverage probability bias-correction, M&S Theorem 4
covbcLD80 <- sum(lobcLD80 < qnorm(alpha80,.5,sigma) &
   qnorm(alpha80,.5,sigma) < upbcLD80)/m

# LD80 coverage probability, Hart
covHLD80 <- sum(HloLD80 < qnorm(alpha80,.5,sigma) &
   qnorm(alpha80,.5,sigma) < HupLD80)/m

#LD80 coverage probability, Hart Bcn
covHbcn2LD80 <- sum(Hbcn2loLD80 < qnorm(alpha80,.5,sigma) &
   qnorm(alpha80,.5,sigma) < Hbcn2upLD80)/m

#LD80 coverage probability, Hart Bnh
```

```
covHbnhLD80 <- sum(HbnhloLD80 < qnorm(alpha80,.5,sigma) &
   qnorm(alpha80,.5,sigma) < HbnhupLD80)/m

# LD80 coverage probability, Hart Bnh asymptotic
covHbnhaLD80 <- sum(HbnhaloLD80 < qnorm(alpha80,.5,sigma) &
   qnorm(alpha80,.5,sigma) < HbnhaupLD80)/m

# cline coverage probability, M&S (5.4)
covcline <- sum(locline < qcline & qcline < upcline)/m

# cline coverage probability, M&S (5.7)
covDQcline <- sum(DQlocline < qcline & qcline < DQupcline)/m

# cline coverage probability bias-correction,M&S Theorem 4
covbccline <- sum(lobccline < qcline & qcline < upbccline)/m

# cline coverage probability, Hart
covHcline <- sum(Hlocline < qcline & qcline < Hupcline)/m

# cline coverage probability, Hart Bcn
covHbcn2cline <- sum(Hbcn2locline < qcline & qcline < Hbcn2upcline)/m

# cline coverage probability, Hart Bnh
covHbnhcline <- sum(Hbnhlocline < qcline & qcline < Hbnhupcline)/m

# cline coverage probability, Hart Bnh asymptotic
covHbnhacline <- sum(Hbnhalocline < qcline & qcline < Hbnhaupcline)/m
```

APPENDIX B

S-PLUS PROGRAMS FOR *lumt* DATA

This Appendix shows the codes for computing LD_{20}, LD_{80}, the cline width, and the corresponding confidence intervals for the *lumt* distances using the *Bcn* method and the Müller and Schmitt (bias-corrected) method. The confidence intervals require the knowledge of the first and second derivatives of the kernel estimator. Our approach for estimating the derivatives is to use some reasonable bandwidth to compute $\hat{p}(x)$ first. With $\hat{p}(x)$, we then estimate $p''(x)$. Using $p''(x)$, we estimate the optimal *h*. Finally, using the optimal *h*, we recalculate $\hat{p}(x)$, $p'(x)$, and $p''(x)$.

Computing an Initial Kernel Estimator and Second Derivative of the Kernel Estimator

This program uses a reasonable bandwidth (0.12) to compute $\hat{p}(x)$ first.

```
n <- length(lumtdata$outcome)
b <- 0.12

# use this bandwidth b to estimate p''(x), then use estimated p''(x) to find optimal b for
    lumt data

yi <- lumtdata$outcome

# transform dist to [0,1] range
lumtdist01 <- lumtdata$dist/max(lumtdata$dist)

den <- density(lumtdist01,n=n,from=0, to=1)
x <- den$x
```

```
f1x <- den$y

mx <- max(diff(x))

gx <- sum(f1x)*mx

fx <- f1x/gx

   xi <- sort(lumtdist01)
   xi1  <-  xi[2:n]
   xi1  <-  c(xi1,1)

   si  <-  (xi+xi1)/2
   si0  <-  c(0,si[1:n-1])

   nx  <-  length(x)

   xk <- -t(outer(x,c(0,si),"-")/b)
   k <- abs(xk)<=1
   int <- t(matrix(0,nx,n+1))
   wint <- matrix(0,n,nx)

   for (j in 1:nx){
     xk <- ifelse(xk<=(-1),-1,xk)
     xk <- ifelse(xk>=1,1,xk)
     int[,j] <- 3/4*((xk[,j]-(xk[,j]^3)/3)-((-1)-((-1)^3)/3))
     wint[,j] <- diff(int[,j])
          }

lumtld20 <- NULL
lumtld80 <- NULL

lumtphat <- rep(0,nx)
lumtp1hatx <- rep(0,nx-1)
lumtp2hatx <- rep(0,nx-2)

lumtpld20 <- NULL
lumtpld80 <- NULL

lumtp1hat20 <- NULL
lumtp2hat20 <- NULL
lumtp1hat80 <- NULL
```

```
lumtp2hat80 <- NULL

        wint <- sweep(wint,2,apply(wint,2,sum),"/")

        lumtphat <- as.vector(yi%*%wint)
        lumtld20 <- approx(lumtphat,x,alpha20)$y
        lumtp1hatx <- diff(lumtphat)/max(diff(x))
        lumtp1hat20 <- approx(x[-1],lumtp1hatx,lumtld20)$y
        lumtp2hatx <- diff(lumtp1hatx)/max(diff(x))
        lumtp2hat20 <- approx(x[-c(1,length(x))],lumtp2hatx,lumtld20)$y

        lumtpld20 <- approx(x,lumtphat,lumtld20)$y

        lumtld80 <- approx(lumtphat,x,alpha80)$y
        lumtp1hat80 <- approx(x[-1],lumtp1hatx,lumtld80)$y
        lumtp2hat80 <- approx(x[-c(1,length(x))],lumtp2hatx,lumtld80)$y

        lumtpld80 <- approx(x,lumtphat,lumtld80)$y

d20 <- approx(x,fx,lumtld20)$y
d80 <- approx(x,fx,lumtld80)$y
```

Estimating First and Second Derivatives of the Kernel Estimator

The following codes compute an optimal bandwidth. Using the optimal bandwidth, we recalculate $\hat{p}(x)$, $p'(x)$, and $p''(x)$.

```
# Compute optimal bandwidth
alpha20 <- 0.2
alpha80 <- 0.8

B <- integrate(function(x){((3/4) * (1 - x^2) * x^2)}/2, -1, 1)$integral
V <- integrate(function(x){((3/4)*(1-x**2))**2}, -1,1)$integral

C120 <- ((alpha20*(1-alpha20)*V)/(d20*4*lumtp2hat20^2*B^2))^.2
b20 <- C120/n^.2

C180 <- ((alpha80*(1-alpha80)*V)/(d80*4*lumtp2hat80^2*B^2))^.2
b80 <- C180/n^.2
```

```
b <- sum(b20,b80)/2

# Recalculate p̂(x), p'(x), and p"(x)

    lumtld20 <- NULL
    lumtld80 <- NULL

    xi <- sort(lumtdist01)

    xi1 <- xi[2:n]
    xi1 <- c(xi1,1)

    si <- (xi+xi1)/2
    si0 <- c(0,si[1:n-1])

    nx <- length(x)

    xk <- -t(outer(x,c(0,si),"-")/b)
    k <- abs(xk)<=1
    int <- t(matrix(0,nx,n+1))
    wint <- matrix(0,n,nx)

    for (j in 1:nx){
      xk <- ifelse(xk<=(-1),-1,xk)
      xk <- ifelse(xk>=1,1,xk)
      int[,j] <- 3/4*((xk[,j]-(xk[,j]^3)/3)-((-1)-((-1)^3)/3))
      wint[,j] <- diff(int[,j])
       }

    lumtphat <- rep(0,nx)
    lumtphatr <- rep(0,nx)
    lumtp1hatx <- rep(0,nx-1)
    lumtp2hatx <- rep(0,nx-2)

    lumtpld20 <- NULL
    lumtpld80 <- NULL

    lumtp1hat20 <- NULL
    lumtp2hat20 <- NULL

    lumtp1hat80 <- NULL
```

```
    lumtp2hat80 <- NULL

        wint <- sweep(wint,2,apply(wint,2,sum),"/")

        lumtphat <- as.vector(yi%*%wint)

        lumtld20 <- approx(lumtphat,x,alpha20)$y
        lumtp1hatx <- diff(lumtphat)/max(diff(x))
        lumtp1hat20 <- approx(x[-1],lumtp1hatx,lumtld20)$y
        lumtp2hatx <- diff(lumtp1hatx)/max(diff(x))
        lumtp2hat20 <- approx(x[-c(1,length(x))],lumtp2hatx,lumtld20)$y

        lumtpld20 <- approx(x,lumtphat,lumtld20)$y

        lumtld80 <- approx(lumtphat,x,alpha80)$y
        lumtp1hat80 <- approx(x[-1],lumtp1hatx,lumtld80)$y
        lumtp2hat80 <- approx(x[-c(1,length(x))],lumtp2hatx,lumtld80)$y

        lumtpld80 <- approx(x,lumtphat,lumtld80)$y

# computing O(1/n) in var(p(x))

d20 <- approx(x,fx,lumtld20)$y
d80 <- approx(x,fx,lumtld80)$y

w1 <- seq(-1,1,.01)
den20 <- density(lumtld20-b20*w1,n=length(w1))

fO20 <- ((3/4)*(1-w1^2))^2/(den20$y)
int120 <- sum(fO20[2:(length(fO20)-1)])*max(diff(w1))
int220 <- V/d20

den80 <- density(lumtld80-b80*w1,n=length(w1))

fO80 <- ((3/4)*(1-w1^2))^2/(den80$y)
int180 <- sum(fO80[2:(length(fO80)-1)])*max(diff(w1))
int280 <- V/d80

lumtf20 <- NULL
lumtO1overnvar20 <- NULL
lumtv20 <- NULL
lumtf80 <- NULL
```

```
lumtO1overnvar80 <- NULL
lumtv80 <- NULL

    lumtf20 <- approx(x,lumtphat,lumtld20)$y
    lumtO1overnvar20 <- (lumtf20*(1-umtf20)/(n*b20))*(int120-int220)
    lumtv20 <- lumtf20*(1-lumtf20)*V/(n*b20*d20) + lumtO1overnvar20

    lumtf80 <- approx(x,lumtphat,lumtld80)$y
    lumtO1overnvar80 <- (lumtf80*(1-umtf80)/(n*b80))*(int180-int280)
    lumtv80 <- lumtf80*(1-lumtf80)*V/(n*b80*d80) + lumtO1overnvar80
```

Computing *Bcn*

```
    Bcn220 <- NULL
    Bcn280 <- NULL

    n <- length(lumtdata$outcome)
    lumtdist01 <- lumtdata$dist/max(lumtdata$dist)

    xi <- sort(lumtdist01)
    xi1 <- xi[2:n]
    xi1 <- c(xi1,1)

    si <- (xi+xi1)/2
    si0 <- c(0,si[1:n-1])

    sig2k <- 2*B
```

compute *Bcn* for LD20

```
        xk120 <- -(lumtld20-c(0, si))/b20
        xk120 <- ifelse(xk120 <= (-1), -1, xk120)
        xk120 <- ifelse(xk120 >= 1, 1, xk120)
        int120 <- (3/4) * ((xk120 - (xk120^3)/3) - ((-1) -  ((-1)^3)/3))
        wint120 <- diff(int120)
        wint1220 <- sum((b20*wint120)^2)

        wh20 <- lumtpld20*(1-lumtpld20)/sum(wint120^2)

        v320 <- lumtpld20*(1-lumtpld20)*wint1220
        Bcn220 <- (C120^3 * n^(-0.6) * 2 * B * lumtp2hat20)/
```

```
    (2 * sqrt(wint1220 * alpha20 * (1-alpha20)))
```

compute *Bcn* for LD80

```
    xk180 <- -(lumtld80-c(0, si))/b80
    xk180 <- ifelse(xk180 <= (-1), -1, xk180)
    xk180 <- ifelse(xk180 >= 1, 1, xk180)
    int180 <- (3/4) * ((xk180 - (xk180^3)/3) - ((-1) -  ((-1)^3)/3))
    wint180 <- diff(int180)
    wint1280 <- sum((b80*wint180)^2)

    wh80 <- lumtpld80*(1-lumtpld80)/sum(wint180^2)

    v380 <- lumtpld80*(1-lumtpld80)*wint1280
    Bcn280 <- (C180^3 * n^(-0.6) * 2 * B * lumtp2hat80)/
       (2 * sqrt(wint1280 * lumtpld80 * (1-lumtpld80)))
```

Computing Covariance of LD20 and LD80 for the *lumt* Distances

```
xi <- sort(lumtdist01)

xi1 <- xi[2:n]
xi1 <- c(xi1,1)

si <- (xi+xi1)/2
si0 <- c(0,si[1:n-1])

nx <- length(x)
pxi <- lumtphat

LD20 <- lumtld20
LD80 <-  lumtld80
```

compute expected value of p(LD20)

```
pLD20 <- lumtpld20
p1LD20 <- lumtp1hat20
p2LD20 <- lumtp2hat20

f20 <- function(u){ifelse(abs((LD20-u)/b)<=1,3/4*(1-abs((LD20-
   u)/b)^2),0)}
```

```
a20 <- rep(0,n)
ag20 <- rep(0,n)
for (i in 1:n){
   a20[i] <-
   integrate(f20,lower=si0[i],upper=si[i],LD20=LD20,b=b)$integral
   ag20[i] <-- pxi[i]*a20[i]
}
EpLD20 <- sum(ag20)/b
VpLD20 <- alpha20*(1-alpha20)*sum(a20^2)/(b^2)

# compute expected value of p(LD80)

pLD80 <- lumtpld80
p1LD80 <- lumtp1hat80
p2LD80 <- lumtp2hat80

# K21: 3/4*(1-x^2)
f80 <- function(u){ifelse(abs((LD80-u)/b)<=1,3/4*(1-abs((LD80-
   u)/b)^2),0)}

a80 <-- rep(0,n)
ag80 <- rep(0,n)
for (i in 1:n){
   a80[i] <- integrate(f80,lower=si0[i],upper=si[i],
   LD80=LD80,b=b)$integral
   ag80[i] <- pxi[i]*a80[i]
}
EpLD80 <- sum(ag80)/b
VpLD80 <- alpha80*(1-alpha80)*sum(a80^2)/(b^2)

# compute expected value of p(LD20) and p(LD80)
eg2080 <- rep(0,n)

for (i in 1:n){
   eg2080[i] <- (pxi[i]*(1-pxi[i])+(pxi[i]^2))*(integrate(f20,lower=si0[i],
   upper=si[i],LD20=LD20,b=b)$integral)*(integrate(f80,
   lower=si0[i],upper=si[i],LD80=LD80,b=b)$integral)
}

cg2080 <- matrix(0,n,n)
for (i in 1:n){
   for (j in 1:n){
```

```
    cg2080[i,j] <- ifelse(i==j,0,(pxi[i]*pxi[j]*(integrate(f20,lower=si0[i],
    upper=si[i],LD20=LD20,b=b)$integral)*(integrate(f80,lower=si0[j],
    upper=si[j],LD80=LD80,b=b)$integral)))
}
    }
Ep20p80 <- (1/b^2)*(sum(eg2080) + sum(cg2080))
Covp20p80 <- Ep20p80 - EpLD20 * EpLD80

# compute cov(LD20,LD80)
covLD20LD80 <- Covp20p80/(p1LD20*p1LD80)

# compute var(LD20) and var(LD80)
varLD20 <- VpLD20/(p1LD20^2)
varLD80 <- VpLD80/(p1LD80^2)

# compute correlation of LD20 and LD80
corLD20LD80 <- covLD20LD80/(sqrt(varLD20)*sqrt(varLD80))
```

Computing 95% Confidence Interval Using *Bcn* Method and M & S (Bias-Corrected) Method

```
beta<-0.5

# compute estimates and CI for lumt using Bcn method
lumtHbcn2LD20 <- NULL
lumtHbcn2loLD20 <- NULL
lumtHbcn2upLD20 <- NULL
lumtlenHbcn2LD20 <- NULL
lumtmpHbcn2LD20 <- NULL

lumtHbcn2LD80 <- NULL
lumtHbcn2loLD80 <- NULL
lumtHbcn2upLD80 <- NULL
lumtlenHbcn2LD80 <- NULL
lumtmpHbcn2LD80 <- NULL

lumtHbcn2cline <- NULL
lumtHbcn2locline <- NULL
lumtHbcn2upcline <- NULL
lumtlenHbcn2cline <- NULL
lumtmpHbcn2cline <- NULL
```

```
# Hart (Bcn)
lumtHbcn2LD20 <- lumtld20 + bcn220*sqrt(lumtv20)/lumtp1hat20
lumtHbcn2loLD20 <- lumtHbcn2LD20 - qnorm(1-beta/2)*
   (sqrt(lumtv20)/sqrt(lumtp1hat20^2))
lumtHbcn2upLD20 <- lumtHbcn2LD20 + qnorm(1-beta/2)*
   (sqrt(lumtv20)/sqrt(lumtp1hat20^2))
lumtlenHbcn2LD20 <- lumtHbcn2upLD20 - lumtHbcn2loLD20
lumtmpHbcn2LD20 <- (lumtHbcn2upLD20 + lumtHbcn2loLD20)/2
lumtHbcn2LD80 <- lumtld80 + bcn280*sqrt(lumtv80)/lumtp1hat80
lumtHbcn2loLD80 <- lumtHbcn2LD80 - qnorm(1-beta/2)*
   (sqrt(lumtv80)/sqrt(lumtp1hat80^2))
lumtHbcn2upLD80 <- lumtHbcn2LD80 + qnorm(1-beta/2)*
   (sqrt(lumtv80)/sqrt(lumtp1hat80^2))
lumtlenHbcn2LD80 <- lumtHbcn2upLD80 - lumtHbcn2loLD80
lumtmpHbcn2LD80 <- (lumtHbcn2upLD80 + lumtHbcn2loLD80)/2

lumtcline <- lumtld20 - lumtld80
lumtHbcn2cline <- lumtHbcn2LD20 - lumtHbcn2LD80
lumtHbcn2locline <- lumtHbcn2cline - qnorm(1-beta/2)*
   sqrt(lumtv20/lumtp1hat20^2 + lumtv80/lumtp1hat80^2 –
   2 *CovLD20LD880)
lumtHbcn2upcline <- lumtHbcn2cline + qnorm(1-beta/2)*
   sqrt(lumtv20/lumtp1hat20^2 + lumtv80/lumtp1hat80^2 –
   2 *CovLD20LD880)
lumtlenHbcn2cline <- lumtHbcn2upcline - lumtHbcn2locline
lumtmpHbcn2cline <- (lumtHbcn2upcline + lumtHbcn2locline)/2

# compute estimates and CI for lumt using M & S bias-corrected method
lumtMSLD20 <- NULL
lumtMSloLD20 <- NULL
lumtMSupLD20 <- NULL
lumtlenMSLD20 <- NULL
lumtmpMSLD20 <- NULL

lumtMSLD80 <- NULL
lumtMSloLD80 <- NULL
lumtMSupLD80 <- NULL
lumtlenMSLD80 <- NULL
lumtmpMSLD80 <- NULL

lumtMScline <- NULL
lumtMSlocline <- NULL
```

```
lumtMSupcline <- NULL
lumtlenMScline <- NULL
lumtmpMScline <- NULL

# bias-corrected CI (Muller and Schmitt Theorem 4)
lumtMSLD20 <- lumtld20 + (b^2*lumtp2hat20*2*B) /(2*lumtp1hat20)
lumtMSsdLD20 <- sqrt((alpha20*(1-alpha20)*V)/(n*b*lumtp1hat20^2))

lumtMSloLD20 <- lumtMSLD20 - qnorm(1-beta/2)*lumtMSsdLD20
lumtMSupLD20 <- lumtMSLD20 + qnorm(1-beta/2)*lumtMSsdLD20
lumtlenMSLD20 <- lumtMSupLD20 - lumtMSloLD20
lumtmpMSLD20 <- (lumtMSupLD20 + lumtMSloLD20)/2

lumtMSLD80 <- lumtld80 + (b^2*lumtp2hat80*2*B) /(2*lumtp1hat80)
lumtMSsdLD80 <-- sqrt((alpha80*(1-alpha80)*V)/(n*b*lumtp1hat80^2))

lumtMSloLD80 <- lumtMSLD80 - qnorm(1-beta/2)*lumtMSsdLD80
lumtMSupLD80 <- lumtMSLD80 + qnorm(1-beta/2)*lumtMSsdLD80
lumtlenMSLD80 <- lumtMSupLD80 - lumtMSloLD80
lumtmpMSLD80 <- (lumtMSupLD80 + lumtMSloLD80)/2

lumtcline <- lumtld20 - lumtld80
lumtMScline <- lumtMSLD20 - lumtMSLD80
lumtMSlocline <- lumtMScline - qnorm(1-beta/2)*
   sqrt((alpha20*(1-alpha20)*V)/(n*b)*
   (1/lumtp1hat80^2+1/lumtp1hat20^2) - 2 * CovLD20LD80)
lumtMSupcline <- lumtMScline + qnorm(1-beta/2)*
   sqrt((alpha20*(1-alpha20)*V)/(n*b)*
   (1/lumtp1hat80^2+1/lumtp1hat20^2) - 2 * CovLD20LD80)
lumtlenMScline <- lumtMSupcline - lumtMSlocline
lumtmpMScline <- (lumtMSupcline + lumtMSlocline)/2
```

REFERENCES

Agresti, A. (2002), *Categorical Data Analysis*, New York: John Wiley and Sons.

Barlow, R. E., Bartholomew, D. J., Bremner, J. M., and Brunk, H. D. (1972), *Statistical Inference Under Order Restrictions*, New York: John Wiley.

Berkson, J. (1944), "Application of the Logistic Function to Bioassay", *Journal of the American Statistical Association, 39*, 357-365.

Bliss, C. I. (1934), "The Method of Probits", *Science*, *79*, 38-39.

Brumfield, R. T., Jernigan, R. W., McDonald, D. B., and Braun, M. J. (2001), "Evolutionary Implications of Divergent Clines in an Avian (*MANACUS*: AVES) Hybrid Zone", *Evolution, 55*(10), 2070-2087.

Cox, D. R., and Snell, E. J. (1970), *Analysis of Binary Data*, London: Chapman & Hall/CRC.

Engeman, R. M., Otis, D. L., and Dusenberry, W. E. (1986), "Small Sample Comparison of Thompson's Estimator to Some Common Bioassay Estimators", *Journal of Statistical Computation and Simulation, 25*, 237-250.

Epanechnikov, V. A. (1969), "Nonparametric Estimates of a Multivariate Probability Density", *Theory of Probability and Its Applications, 14*, 153-158.

Filliben, J. J. (1975), "The Probability Plot Correlation Coefficient Test for Normality", *Technometrics*, *17*(1), 111-117.

Finney, D. J. (1978), *Statistical Method in Biological Assay*, London: Charles Griffin & Company Ltd.

Gasser, Th., and Müller, H.-G. (1979), "Kernel Estimation of Regression Functions", *Smoothing Techniques for Curve Estimation*, Springer Lecture Notes in Mathematics No. 757, Springer-Verlag, Berlin, 23-68.

Gelman, A., Pasarica, C., and Dodhia, R. (2002), "Statistical Computing and Graphics Let's Practice What We Preach: Turning Tables into Graphs", *The American Statistician, 56*(2), 121-130.

Green, P. J., and Silverman, B. W. (1994), *Nonparametric Regression and Generalized Linear Models*, Chapman & Hall.

Hall, P., and Muller, H. G. (2003), "Order-Preserving Nonparametric Regression, with Applications to Conditional Distribution and Quantile Function Estimation", *Journal of American Statistical Association, 98*(463), 598-608.

Hamilton, M. A. (1979), "Robust Estimates of the ED 50", *Journal of the American Statistical Association*, *74*, 344-354.

Hardle, W. (1990), *Applied Nonparametric Regression*, Cambridge University Press.

Hart, J. D. (1997), *Nonparametric Smoothing and Lack-of-Fit Tests*, New York: Springer-Verlag.

Huber, P. J. (1972), "The 1972 Wald Lecture Robust Statistics: A Review", *The Annals of Mathematical Statistics*, *43*(4), 1041-1067.

Jaarola, M., Tegelstrom, H., and Fredga, K. (1997), "A Contact Zone with Noncoincident Clines for Sex-Specific Markers in the Field Vole (Microtus Agresti)", *Evolution, 51*(1), 241-249.

James, B. R., James, K. L., and Westenberger, H. (1984), "An Efficient *R*-Estimator for the ED50", *Journal of the American Statistical Association, 79*(385), 164-173.

Kelly, G. E. (2001), "The Median Lethal Dose – Design and Estimation", *The Statistician, 50*(1), 41-50.

McCullagh, P., and Nelder, J. A. (1989), *Generalized Linear Models*, London: Chapman and Hall.

Müller, H.-G., and Schmitt, T. (1988), "Kernel and Probit Estimates in Quantal Bioassay", *Journal of the American Statistical Association, 83*(403), 750-759.

Miller, R. G., and Halpern, J. W. (1980), "Robust Estimators for Quantal Bioassay", *Biometrika*, *67*, 103-110.

Nadaraya, E. A. (1964), "On Estimating Regression", *Theory of Probability and Its Applications*, *9*, 141-142.

Spearman, C. (1908), "The Method of Right and Wrong Cases (Constant Stimuli) Without Gauss's Formulae", *British Journal of Psychology*, *2*, 227-242.

Thompson, W. R. (1947), "Use of Moving Averages and Interpolation to Estimate Median Effective Dose, I: Fundamental Formulas, Estimation of Error, and Relation to Other Methods", *Bacteriological Reviews*, *11*, 116-145.

Vogel, R. M. (1986), “The Probability Plot Correlation Coefficient Test for the Normal, Lognormal, and Gumbel Distributional Hypotheses”, *Water Resources Research*, *22*(4), 587-590.

Watson, G. S. (1964), “Smooth Regression Analysis”, *Sankhya*, *26*, 359-372.