7-18-2008

New Non-Parametric Confidence Interval for the Youden

Haochuan Zhou

NEW NON-PARAMETRIC CONFIDENCE INTERVAL FOR THE YOUDEN INDEX

by

Haochuan Zhou

Under the direction of Gengsheng Qin

ABSTRACT

Youden index, a main summary index for the Receiver Operating Characteristic (ROC) curve, is a comprehensive measurement for the effectiveness of a diagnostic test. For a continuous-scale diagnostic test, the optimal cut-point for the positive of disease is the cut-point leading to the maximization of the sum of sensitivity and specificity. Finding the Youden index of the test is equivalent to maximize the sum of sensitivity and specificity for all the possible values of the cut-point. In this thesis, we propose a new non-parametric confidence interval for the Youden index. Extensive simulation studies are conducted to compare the relative performance of the new interval with the existing intervals for the index. Our simulation results indicate that the newly developed non-parametric method performs as well as the existing parametric method but it has better finite sample performance than the existing non-parametric methods. The new method is flexible and easy to implement in practice. A real example is also used to illustrate the application of the proposed interval.

KEY WORDS: Confidence intervals; Optimal cut-point; ROC curve; Sensitivity; Specificity; Youden index.

NEW NON-PARAMETRIC CONFIDENCE INTERVAL FOR THE YOUDEN INDEX

by

Haochuan Zhou

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

In the College of Arts and Sciences

Georgia State University

2008

NEW NON-PARAMETRIC CONFIDENCE INTERVAL FOR THE YOUDEN INDEX

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

In the College of Arts and Sciences

Georgia State University

2008

by

Haochuan Zhou

Major Professor: Dr. Gengsheng Qin
Committee: Dr. Yu-sheng Hsu
Dr. Yixin Fang

Electronic Version Approval:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
August 2008

ACKNOWLEDGEMENTS

I would like to acknowledge those who have helped me in the completion of this thesis.

First of all, I like to thank Dr. Qin who is an extraordinary advisor, a great professor, and a great person. There is no way I could have finished this thesis without his generous support. I have learned tremendously under Dr. Qin's guidance on this thesis and also by attending his classes in different semesters during the past two years. He has always been available whenever I had any questions.

I would also like to acknowledge the other members of my thesis committee, Dr.Yu-Sheng Hsu, and Dr. Yixin Fang for taking the time to read this thesis and provide useful comments. I have attended at least one class taught by two of the above mentioned professors and have found all of the classes taught by them very helpful.

I'd also like to thank all the other professors in the Mathematics & Statistics Department at Georgia State University for their support and guidance. It would not have been possible for me to complete the requirements of the graduate program without their guidance. In addition, I thank my classmates Meng Zhao, Xin Huang, Fangfang Sun, and Baoying Yang for their help and discussion.

Lastly, I want to acknowledge the support of my parents, who have always been there for me with words of encouragement and wisdom.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER I

INTRODUCTION

Over the last thirty years, the Receiver Operating Characteristic (ROC) curve analysis has obtained regards in medical fields as a statistical method to evaluate the discriminating efficiency of diagnostic tests. The ROC curve describes the ability of a test to suitably diagnose for a variety of test cutoff points (Pepe, 2003). The ROC curve supplies an easily graphical way for statisticians to compare the efficiency at special levels of false negative rates and optimal diagnostic ability of different tests.

For a continuous-scale diagnostic test, the optimal cut-point for the positive test result of disease is the cut-point leading to the maximization of the sum of sensitivity and specificity (Schisterman et al., 2007). The cut-point determination by this maximization procedure is equivalent to minimization of the sum of false negative and false positive misclassification likelihoods. The cut-points obtained by this method certainly have the clinically desirable property of maximizing the overall correct diagnosis rate and therefore minimizing the overall misdiagnosis rate (Kim, 2008).

Let c_0 be the optimal cut-point of test results, Youden (1950) introduced the following index for the ROC curve:

$$J = sensitivity(c_0) + specificity(c_0) - 1.$$

So, finding the optimal cut-point is equivalent to estimate the Youden index J. This index is an important summary measure for the ROC curve. Youden indicated that the index J has several desirable features. For example, the possible range of values for J is from zero to one with the value zero stating a totally useless test and the value one indicating an ideal diagnostic test; the index is independent of the relative sizes of the

control (non-diseased) and diseased groups; it is also independent of the absolute sizes of the control and diseased groups; All the tests that have the same index make the same total number of misclassifications per hundred patients. Graphically, the Youden index is the maximum vertical distance between the ROC curve and the diagonal chance line, and plays as a comprehensive measurement of the optimal clinical diagnostic ability. The Youden index has found many applications in medical and biological sciences. For instances, Demir et al. (2002) applied it to identify the most reliable indices in differentiation between thalassemial trait and iron deficiency anemia, and Schisterman et al. (2007) used it to analyze a dataset on the Coronary Calcium Score, a marker for atherosclerosis.

Youden index is a function of sensitivity and specificity that depend on the underlying distributions for the diseased and non-diseased populations. Under the assumption that the distributions belong to a specific parametric family such as binormal distributions, Fluss et al. (2005) and Schisterman et al. (2007) provided statistical inference for the index. However, the parametric methods, which assume parametric models for both the diseased and non-diseased population, may be sensitive to departures from the distributional assumptions and can only, provided a limited range of distributional forms. Therefore, a well performed non-parametric method for the inference of the Youden index is much needed. A few studies (e.g., Barkan, 2001; Fluss et al., 2005) have considered constructing non-parametric confidence intervals for the Youden index and the corresponding cutoff point. All these non-parametric approaches used the empirical distributions as the estimates of the unknown underlying distributions

for the diseased and non-diseased populations and makes inference for the index based on normal approximation and bootstrap methods.

In this thesis, we focus on construction of non-parametric confidence interval for the Youden index. A new non-parametric interval for the index is proposed based on the bootstrap variance estimate of the empirical index estimate. The new interval could be regarded as an extension of Agresti and Coull's interval (1998) for a binomial proportion. Extensive simulation studies are conducted to compare the relative performance of the new interval with the existing intervals for the index. Our simulation results indicate that the newly developed non-parametric method performs as well as the existing parametric method but it has better finite sample performance than the existing bootstrap percentile (BP) and BCa methods. The new method is flexible and easy to implement in practice.

This thesis is organized as follows. In Chapter II, we give the empirical estimate for the Youden index. In Chapter III, we describe the existing methods for interval estimation of the Youden Index. In Chapter IV, we propose our new method for construction of non-parametric confidence interval of the Youden index. In Chapter V, we conduct simulation studies to assess the finite-sample performance of the new interval and the existing intervals. A real example is used to illustrate the application of the proposed interval in Chapter VI. In Chapter VII, we conclude with a short discussion.

CHAPTER II

EMPIRICAL ESTIMATION FOR THE YOUDEN INDEX

The **sensitivity** of a diagnostic test is defined as the probability of correctly classifying diseased patients.

The **specificity** of a diagnostic test is defined as the probability of correctly classifying non-diseased subjects.

Let X and Y be results of a continuous-scale test for a diseased and non-diseased subject, respectively. Without loss of generality, we assume that X and Y are independent. For a given cut-off point c, we can define sensitivity and specificity of the test as

$$sensitivity(c) = P(X \geq c),\ specificity(c) = P(Y \leq c)$$

respectively. Let F and G be the respective distribution functions of X and Y. We can then write $sensitivity(c) = 1 - F(c)$ and $specificity(c) = G(c)$. It is evident that there exists a reversing relationship between sensitivity and specificity; moving the cutoff point enlarges one and meanwhile decreases the other. The Youden Index is defines as

$$\begin{aligned} J &= \max\{sensitivity(c) + specificity(c) - 1\} \\ &= \max\{P(X \geq c) + P(Y \leq c) - 1\} \end{aligned}$$

Let c_o be the cut-off point that maximizes the sum of sensitivity and specificity, we can rewrite the index J as

$$J = sensitivity(c_o) + specificity(c_o) - 1 = P(X \geq c_o) + P(Y \leq c_o) - 1$$

Let $x_1, x_2, \ldots, x_n$ be the test results from *n* diseased patients and $y_1, y_2, \ldots, y_m$ the test results from *m* non-diseased subjects. For a given empirical estimate $\hat{c}_o$ of the optimal cut-off point, the test results can be classified into the table below

		Disease Status		
		Diseased	Non-Diseased	
Diagnostic Test Result	+	**True positive (*A*)**	**False positive (*B*)**	***A+B***
	-	**False negative (*C*)**	**True negative (*D*)**	***C+D***
		A+C=n	***B+D=m***	***n+m***

Table 1

where

$A = \sum_{i=1}^{n} I\left(x_i \geq \hat{c}_o\right)$, and $D = \sum_{j=1}^{m} I\left(y_j \leq \hat{c}_o\right)$, $I(\cdot)$ is the indicator function.

When the distribution functions F and G are known, the sensitivity and specificity at $\hat{c}_o$ can be found by

$$Sensitivity = 1 - F\left(\hat{c}_o\right), \quad Specificity = G\left(\hat{c}_o\right).$$

Then the Youden Index can be calculate as

$$\hat{J} = G(\hat{c}_0) - F(\hat{c}_0).$$

Without knowing the underlying distributions of the diseased and non-diseased populations, the sensitivity and specificity can be estimated by

$$Sensitivity = \frac{A}{A+C} = \frac{\sum_{i=1}^{n} I\left(x_i \geq \hat{c}_o\right)}{n}, \qquad specificity = \frac{D}{B+D} = \frac{\sum_{j=1}^{m} I\left(y_j \leq \hat{c}_o\right)}{m}.$$

Then the empirical estimate for the Youden index is given by

$$\hat{J} = \frac{\sum_{i=1}^{n} I\left(x_i \geq \hat{c}_0\right)}{n} + \frac{\sum_{j=1}^{m} I\left(y_j \leq \hat{c}_0\right)}{m} - 1 = \frac{\sum_{j=1}^{m} I\left(y_j \leq \hat{c}_0\right)}{m} - \frac{\sum_{i=1}^{n} I\left(x_i \leq \hat{c}_0\right)}{n}$$

CHAPTER III

EXISTING CONFIDENCE INTERVALS FOR YOUDEN INDEX

In this chapter, we consider a diagnostic test which yields continuous measurements and is performed on *n* diseased patients and *m* non-diseased subjects. Let $X_1, X_2, \ldots, X_n$ be the test results of a random sample of diseased patients, and $Y_1, Y_2, \ldots, Y_m$ the test results of a random sample of non-diseased subjects. Based on these observations, we attempt to construct $(1-\alpha)100$ percent confidence intervals for the Youden index.

1. Parametric confidence intervals

Parametric method is an efficient method to make inference about the Youden index when the underlying distributions for the diseased and non-diseased populations belong to a specific parametric family.

Schisterman and Perkins (2007) considered binormal distributions as the distributions for the test results from the diseased and non-diseased subjects. They made the following normality assumption about *X* and *Y*:

$$X_i \sim N\left(\mu_x, \sigma_x^2\right),\ i = 1,2,.....n \text{ and } Y_j \sim N\left(\mu_y, \sigma_y^2\right),\ j = 1,2,.....m .$$

Under this assumption, the optimal cut-point leading to *J* occurs at one of the intersections of the two normal probability density functions. It has the following closed form:

$$c_o = \frac{\mu_Y\left(b^2-1\right)-a \pm b\sqrt{a^2+\left(b^2-1\right)\sigma_Y^2 \ln\left(b^2\right)}}{\left(b^2-1\right)}$$

where $a = \mu_X - \mu_Y$ and $b = \sigma_X / \sigma_Y$.

When the two populations have the same variance, the optimal cut-off point c_o is simply the midpoint between the means,

$$c_o = \frac{\mu_X + \mu_Y}{2}$$

In this case, the Youden index is

$$J = \Phi\left(\frac{\mu_x - c_o}{\sigma_x}\right) + \Phi\left(\frac{c_o - \mu_y}{\sigma_y}\right) - 1$$

where Φ denotes the standard normal cumulative distribution function.

Let $\hat{J}$ and $\hat{c}_o$ be the estimates for J and c_o by substituting for the unknown parameter $\mu_X, \mu_Y, \sigma_X, \sigma_Y$ in formulas above their corresponding sample means and standard deviations. Using Delta method, Schisterman and Perkins (2007) obtained the approximate estimate, $Var\left(\hat{J}\right)$ for the variance of the estimated Youden index. Then a normal approximation based confidence intervals (called Delta interval) for J is:

$$\left(\hat{J} - z_{1-\alpha/2}\sqrt{Var\left(\hat{J}\right)}, \hat{J} + z_{1-\alpha/2}\sqrt{Var\left(\hat{J}\right)}\right)$$

where $z_{1-\alpha/2}$ is the $\left(1-\alpha/2\right)$-th quantile of the standard normal distribution.

Schisterman and Perkins (2007) also assumed the following Gamma distributions as the underlying distributions for the diseased and non-diseased subjects:

$$X_i \sim \Gamma\left(\alpha_x, \beta_x\right),\ i = 1,2,.....n \text{ , and } Y_j \sim \Gamma\left(\alpha_y, \beta_y\right),\ j = 1,2,.....m .$$

In this situation, the optimal cut-off point c_o does not have a closed form. We have to numerically find c_o by solving $f(c) \approx g(c)$, where *f* and *g* are the respective probability density functions of $\Gamma(\alpha_x, \beta_x)$ and $\Gamma(\alpha_y, \beta_y)$. The Youden index is now

$$\begin{aligned} J &= G(c_o) - F(c_o) \\ &= \int_0^{c_o} \frac{e^{-y/\beta_y} y^{\alpha_y - 1}}{\beta_y^{\alpha_y} \Gamma(\alpha_y)} dy - \int_0^{c_o} \frac{e^{-x/\beta_x} x^{\alpha_x - 1}}{\beta_x^{\alpha_x} \Gamma(\alpha_x)} dx. \end{aligned}$$

After obtaining the empirical estimate $\hat{J}$ for the index *J*, they derived the approximate variance of $\hat{J}$ by Delta method and constructed a normal approximation based confidence intervals for J .

2. Bootstrap Confidence Intervals

Although the variances of $\hat{J}$ can be obtained by using Delta methods for the parametric approaches mentioned above, they have mathematically complex expression which limit the applications of the parametric confidence intervals in practice. Therefore, Schisterman and Perkins (2007) also suggested two non-parametric confidence intervals for the index by using the bootstrap technique.

The first non-parametric interval for the index is the usual bootstrap percentiles (BP) interval. Note that the empirical estimate for the index *J* is

$$\hat{J} = \frac{\sum_{j=1}^{m} I\left(Y_j \le \hat{c}_o\right)}{m} - \frac{\sum_{i=1}^{n} I\left(X_i \le \hat{c}_o\right)}{n}$$

where $\hat{c}_o$ is the estimate for the optimal cut-off point c_o. In order to construct the BP interval, we can repeatedly draw bootstrap resample from the original sample and

calculate B (≥ 200) bootstrap copies $\left\{J_1^*, J_2^*, ..., J_B^*\right\}$of $\hat{J}$, the lower bound and upper bound of the $(1-\alpha)100\%$ BP interval for the index are simply the $\frac{\alpha}{2}-th$ and $(1-\frac{\alpha}{2})-th$ quantiles of $\left\{J_1^*, J_2^*, ..., J_B^*\right\}$ respectively.

The second non-parametric interval for the index they recommended is the bias corrected and acceleration (BCa) bootstrap interval. Details of this method were laid out by Carpenter and Bithell (2000), and Schisterman and Perkins (2007).

CHAPTER IV

NEW NON-PARAMETRIC CONFIDENCE INTERVAL

Bootstrap is a commonly used method for constructing confidence intervals of unknown parameter when the sampling distribution of the estimator for the parameter is cumbersome or unknown. The BCa interval for the index suggested by Schisterman and Perkins (2007) has acceptable coverage probabilities. However, as shown in this thesis, it can still have poor coverage accuracy in many circumstances. This motivates us to search a better confidence interval for the Youden index.

Let

$$V_i = I\left(X_i < c_o\right), i = 1,2,...,n \text{ and } W_j = I\left(Y_j \le c_o\right), j = 1,2,...,m.$$

Then, $\sum_{i=1}^{n} V_i$ and $\sum_{j=1}^{m} W_j$ are two binomial random variables with proportions $P\left(X < c_o\right)$ and $P\left(Y \le c_o\right)$ respectively. The Youden index,

$$J = P\left(X \ge c_o\right) + P\left(Y \le c_o\right) - 1 = P\left(Y \le c_o\right) - P\left(X < c_o\right),$$

is the difference between two binomial proportions. For a given estimate $\hat{c}_o$ of the optimal cut-off point, the empirical estimate,

$$\hat{J} = \frac{\sum_{j=1}^{m} I\left(Y_j \le \hat{c}_o\right)}{m} - \frac{\sum_{i=1}^{n} I\left(X_i < \hat{c}_o\right)}{n}$$

for the Youden index is the estimated difference between two binomial proportions. Since $\hat{c}_o$ is an empirical estimate for the optimal cut-off point based on original samples, and the indicator variables $I\left(X_i < \hat{c}_o\right), i = 1,2,...,n$ and $I\left(Y_j \le \hat{c}_o\right), j = 1,2,...,m$ are not

independent, $\hat{J}$ is no longer the difference between two simple binomial proportions. Therefore, the usual methods for construction of confidence interval for the difference between two binomial proportions cannot be directly applicable here. Inspired by Agresti and Coull's interval (called AC interval, 1998) for a binomial proportion, and Zhou and Qin's (2003) successful application for construction of confidence intervals of sensitivity, we propose the following procedure for the construction of confidence intervals of the Youden index.

Let

$$\hat{J}_{AC} = \frac{\sum_{j=1}^{m} I\left(Y_j \le \hat{c}_o\right) + z_{1-\alpha/2}^2/2}{m + z_{1-\alpha/2}^2} - \frac{\sum_{i=1}^{n} I\left(X_i < \hat{c}_o\right) + z_{1-\alpha/2}^2/2}{n + z_{1-\alpha/2}^2}$$

Since $z_{1-\alpha/2}$ is approximately equal to 2 when $\alpha = 0.05$, $\hat{J}_{AC}$ may be regarded as an adjusted estimate for the difference between two binomial proportions by adding two successes and two failures to the Bernoulli observations. $\hat{J}_{AC}$ is also the difference between two correlated proportions. Hence, it is difficult to find the variance of $\hat{J}_{AC}$. Therefore, the most often used Wald interval cannot be directly applicable here. In this thesis we propose to use bootstrap method to estimate the variance of $\hat{J}_{AC}$. We summarize the procedure for computing the bootstrap variance in the following steps:

1. From the original data, empirically locate the optimal cut-off point $\hat{c}_o$ which maximizes $\hat{J}$ meanwhile gives us the largest sensitivity. Schisterman and Perkins (2007) mentioned that there may be multiple solutions for $\hat{c}_o$ with the

maximization of $\hat{J}$. The reasonable choice of $\hat{c}_o$ is the one which returns the largest sensitivity value.

2. Draw a bootstrap sample of size n, X_i^*'s, with replacement from the diseased sample X_i's, and a separate bootstrap sample of size m, Y_j^*'s, with replacement from the non-diseased sample Y_j's.

3. Calculate the bootstrap version of $\hat{J}_{AC}$

$$\hat{J}_{AC}^* = \frac{\sum_{j=1}^{m} I\left(Y_j^* \le \hat{c}_0^*\right) + z_{1-\alpha/2}^2/2}{m + z_{1-\alpha/2}^2} - \frac{\sum_{i=1}^{n} I\left(X_i^* \le \hat{c}_0^*\right) + z_{1-\alpha/2}^2/2}{n + z_{1-\alpha/2}^2}$$

4. Repeat step 2 and step 3 B times to obtain the set of bootstrap replications $\left\{\hat{J}_{AC}^{*b} : b = 1,2,.....,B\right\}$ (It is recommended that $B \ge 200$, here we choose B=300). Then, the bootstrap variance estimator $V^*\left(\hat{J}_{AC}\right)$ is defined by

$$V^*\left(\hat{J}_{AC}\right) = \frac{1}{B-1}\sum_{b=1}^{B}\left(J_{AC}^{*b} - \bar{J}_{AC}^*\right)^2$$

where $\bar{J}_{AC}^* = \frac{1}{B}\sum_{b=1}^{B}\hat{J}_{AC}^{*b}$.

Using this bootstrap variance estimator, we propose the following $(1-\alpha)-th$ confidence interval (called NB interval) for the Youden index:

$$\left(\bar{J}_{AC}^* - z_{1-\alpha/2}\sqrt{V^*\left(\hat{J}_{AC}\right)}, \bar{J}_{AC}^* + z_{1-\alpha/2}\sqrt{V^*\left(\hat{J}_{AC}\right)}\right).$$

CHAPTER V

Simulation Study

In this chapter, we conduct two simulation studies to compare the coverage accuracy and interval length of the newly proposed non-parametric (NB) interval with those of the existing parametric (Delta) interval and bootstrap (BP and BCa) intervals for the Youden index J in finite-sample sizes. In both simulation studies, we generated 3000 random samples of size n from the distribution function F for test responses of diseased patients and another independent random sample of size m from the distribution function G for test responses of non-diseased subjects. The sample sizes (n, m) are chosen to be (50, 50), (80, 80), (50, 80), (100, 100), (100, 150), (200, 200), respectively.

In the first study, binormal distributions are chosen to be the underlying distributions. Specifically, we choose

$$X \sim F = N\left(\mu_x, \sigma_x^2\right), \text{ and } Y \sim G = N(0,1),$$

where the variances σ_x^2 is set to be 0.5, 1, 3, and 5 respectively, the μ_x is chosen such that the desired levels of the Youden J = 0.4, 0.6, 0.8, 0.9 are achieved. Table 1.1 gives the detailed parameter setting in this simulation study.

In the second study, we choose Gamma distributions to be the underlying distributions. i.e,

$$X \sim F = \Gamma\left(\alpha_x, \beta_x\right), \text{ and } Y \sim G = \Gamma\left(\alpha_y, \beta_y\right),$$

where $\alpha_y = 1.5$, $\beta_y = 1$, and α_x is set to be 1.5, 2, 2.5, and 3 respectively, β_x is chosen such that the Youden *J* equals 0.4, 0.6, 0.8, and 0.9 respectively. The detailed parameter setting in the study is displayed in Table 1.2.

In both studies, we choose *B* = 300. Under these simulation settings, we generate simulated datasets and calculate the 95% NB, BP, BCa and Delta intervals for the Youden index. The coverage probabilities and average interval lengths are reported in the Tables 2.1.2 – 3.2.6 in Appendix. From these tables, we observe that the proposed NB intervals and the existing Delta intervals have similar coverage accuracy. But the NB intervals perform better than the existing BCa intervals. The BP intervals have the worst performance among the four intervals.

CHAPTER VI

Real Application

In this chapter, we use the data set from Le (2006) to illustrate the application of the proposed interval for the Youden index.

Prostate cancer is a common malignancy in men. After the prostate cancer being confirmed, the next concern for the patient is whether the cancer has spread to the neighboring lymph nodes. The detection of the spread of the cancer is very important in patient management because complete cure is more likely for patients in the early stage of prostate cancer. The spread of prostate cancer could be found by some strategy such as some kind of surgery. As surgery involves big risks for some patients, instead, a new method with extremely small risk is used by testing the "acid phosphatase level in blood serum". So it is of interest to know if the test results can predict the nodal involvement. How accurate is the testing method in detecting the spread of prostate cancer?

In the study, there were 53 patients with prostate cancer. 20 of them are with nodal involvement, and others are not. The data set of the acid phosphatase levels in blood serum is shown as below:

Patient without nodal involvement: 40, 40, 46, 47, 48, 48, 49, 49, 50, 50, 50, 50, 50, 52, 52, 55, 55, 56, 59, 62, 62, 63, 65, 66, 71, 75, 76, 78, 83, 95, 98, 102, 187.

Patient with nodal involvement: 48, 49, 51, 56, 67, 67, 67, 70, 70, 72, 76, 78, 81, 82, 82, 84, 89, 99, 126, 136.

First, we want to know if we can use parametric method to construct confidence interval for the Youden index. The following graphs are the QQ-plots of the data set:

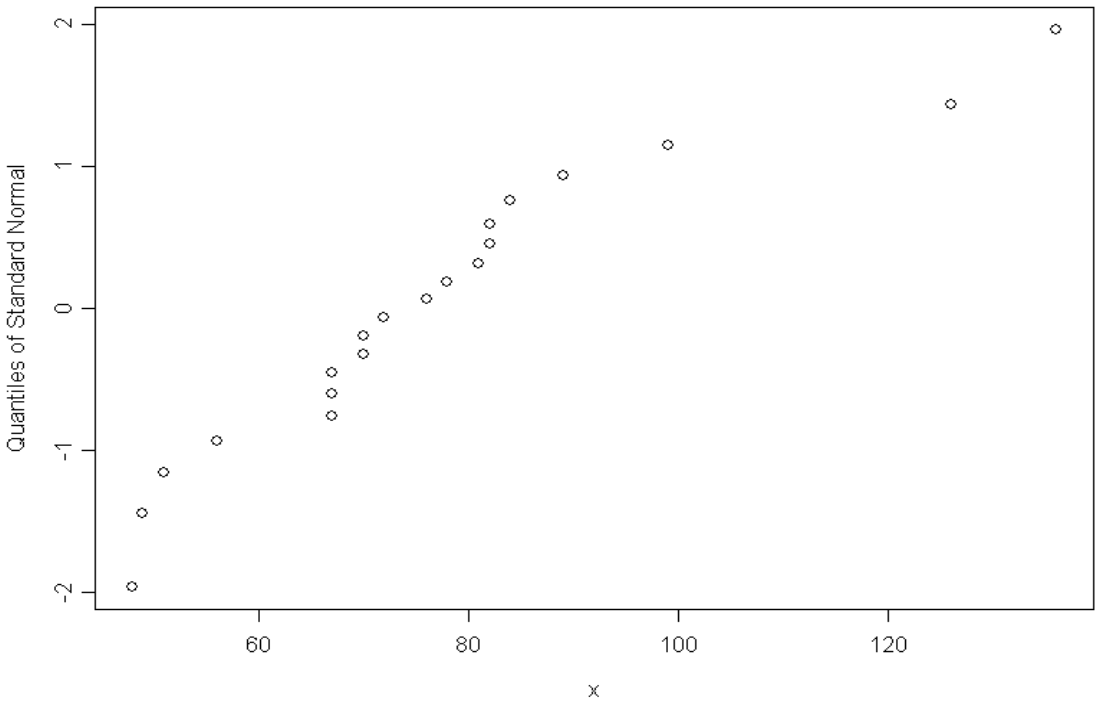

Figure 7.1

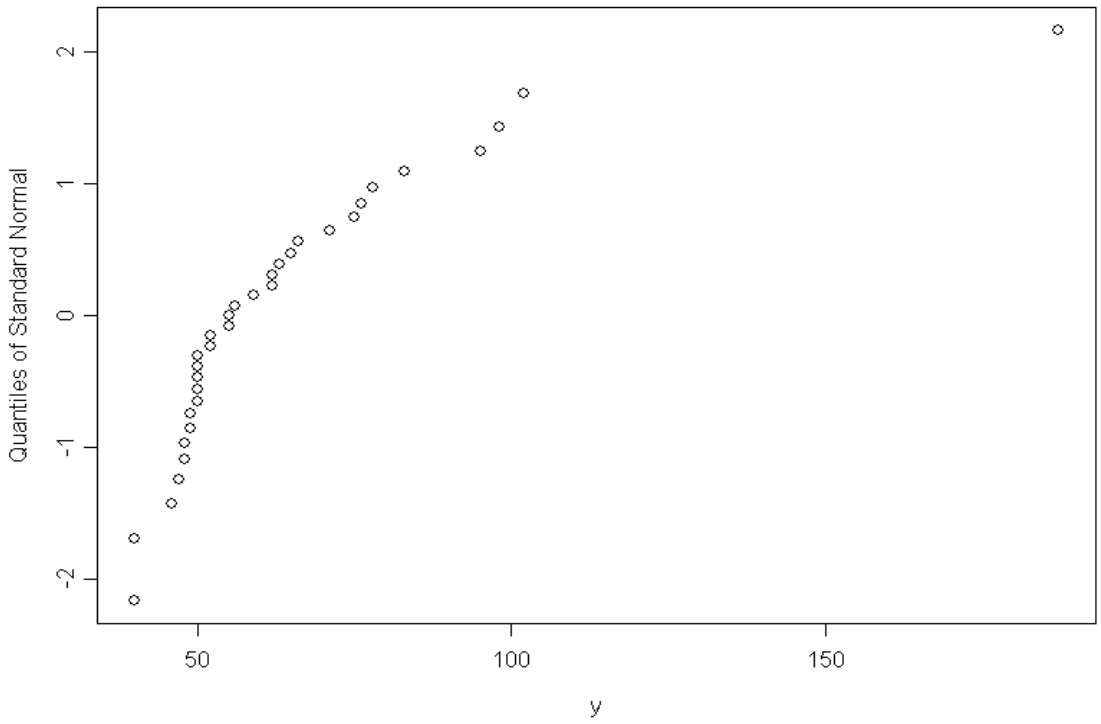

Figure 7.2

From the QQ-plots, we observe that the binormal distribution assumption is not appropriate for both diseased and non-diseased people. It seems that the parametric method cannot be directly applied here. So we try to use non-parametric methods to construct confidence intervals for the Youden index. Using the newly proposed method, we obtain the 95% NB interval:

$$(0.24, 0.67).$$

By using the nonparametric methods suggested by Schisterman and Perkins (2007), we obtain the 95% BCa and BP intervals for the Youden index as follows:

$$\text{BCa interval: } (0.26, 0.74); \text{ BP intervals: } (0.26, 0.75).$$

Le (2006) gave an estimate for the Youden index which is 0.29. This estimated value of the Youden index falls in all the three intervals. Since the NB interval has the

shortest interval length among the three intervals, by our simulation study, we would like to use $(0.24, 0.67)$ as the confidence interval for the Youden index. So, the new testing method has low to moderate accuracy in detecting the spread of prostate cancer.

CHAPTER VII

Discussion and Conclusion

The Youden index is a frequently used summary measure of the ROC curve. It provides a criterion for evaluating the optimal threshold value of a test for which the sum of sensitivity and specificity is maximized. The Youden index is easy to compute and understand. It has found many applications in different fields such as medical informatics and bioinformatics. Many of these applications simply involve the point estimation of the index. Few of them give the interval estimation for the index. However, providing a reliable range for the index is more important than giving a single estimated value for the unknown index in practice. In this thesis, we provide a simple and well performed non-parametric interval estimation method for the index. We have compared the new (NB) interval with the existing parametric (Delta) and non-parametric (BCa and BP) intervals through extensive simulation studies. The NB interval and the Delta interval have similar coverage probabilities in most cases considered here. When the Youden index is high (J = 0.90, both the sensitivity and the specificity must be at least at level of 0.90), the Delta interval performs better than the NB interval when sample sizes are small ((n, m) = (50, 50), (50, 80)). This is not surprising because the NB interval is purely a non-parametric approach and makes no assumptions about the distributional forms for diseased and non-diseased populations. However, the validity of the Delta interval is relied on the parametric distributional assumptions. Comparing with the BCa and BP intervals, the new NB interval performs the best and the BP interval has the worst performance in terms of coverage probability among the three intervals. The BCa interval may have good

coverage probabilities in some cases but it is computationally the most extensive method among the three methods. The BCa intervals always have longer average interval length than the NB intervals.

In summary, we suggest the use of the newly developed NB interval for the Youden index when the underlying distributions for diseased and non-diseased populations are unknown. The Delta interval can be used when the underlying parametric models are valid.

REFERENCES

Agresti, A, Coull B.A., (1998) Approximate is better than 'exact' for interval estimation of binomial proportions. *American Statistician*; **52**: 119-126

Carpenter, J., Bithell, J. (2000). Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. *Statistics in Medicine*; **19**: 1141-1164

Demir, A., Yarali, N., Fisgin, T., Duru, F., and Kara, A. (2002). Most reliable indices in differentiation between thalassemial trait and iron deficiency anemia. *Pediatrics International.* **44:** 612-616.

Fluss, R., Faraggi, D., Reiser, B. (2005). Estimation of the Youden index and its associated cutoff point. *Biometrical Journal*; **47**: 458-472

Kim, K. (2008). Maximization of the sum of sensitivity and specificity as a diagnostic cutpoint criterion. Journal of Clinical Epidemiology. **61**: 516-518.

Le, C.T. (2006). A solution for the most basic optimization problem associated with ROC curve. *Statistical Methods in Medical Research*; **15**: 571-584

Miller R.G., Efron, B., Brown, Jr B.W., Moses, L.E., (1980) *Biostatistics casebook*; Wiley-Interscience; 3-18

Pepe, M.S. (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction*. New York: Oxford University Press

Perkins, N., Schisterman, E.F., (2005). The Youden Index and the optimal cut-point corrected for measurement error. *Biometrical Journal*; **47**: 428-441

Platt, R.W., Hanley, J.A., Yang, H. (2000). Bootstrap confidence intervals for the sensitivity of a quantitative diagnostic test. *Statistics in Medicine*; 19: 313-322

Schisterman, E.F., Faraggi, D., Reiser, B., and Hu, J. (2007). Youden Index and the optimal threshold for markers with mass at zero. *Statistics in Medicine* (in press). Doi: 10.1002/sim. 2993.

Schisterman, E.F., Perkins, N. (2007). Confidence interval for Youden index and corresponding optimal cut-point. *Communication in Statisitics-Simulation and Computation*; **36**: 549-563

Shapiro, D.E., (1998). The interpretation of diagnostic tests. *Statistical Methods in Medical Reasearch*; **8**: 113-134

Youden, W.J., (1950). Index for rating diagnostic tests. *Cancer*; **3**: 32-35

Zhou, X.H., Qin, G.S. (2005). Improved confidence intervals for the sensitivity at a fixed level of specificity of a continuous-scale diagnostic test. *Statistical in Medicine*; **24**: 465-477

APPENDIX

APPENDIX A: Simulation Tables

1. Parameter table in simulation

Youden Index	Standard deviation of disease population			
	$\sigma_x^2 = 0.5$	$\sigma_x^2 = 1$	$\sigma_x^2 = 3$	$\sigma_x^2 = 5$
$J = 0.4$	0.8484	1.0489	1.2540	1.2815
$J = 0.6$	1.4071	1.6833	2.1843	2.4493
$J = 0.8$	2.1682	2.5632	3.4247	3.9625
$J = 0.9$	2.7927	3.2898	4.4340	5.1777

Table 1.1 Normal Distribution: Mean parameter μ_x of diseased group

Youden Index	Shape parameter of disease population			
	$\alpha_x = 1.5$	$\alpha_x = 2$	$\alpha_x = 2.5$	$\alpha_x = 3$
$J = 0.4$	0.4028	0.6016	0.8189	1.0522
$J = 0.6$	0.2295	0.3617	0.5064	0.6614
$J = 0.8$	0.1022	0.1769	0.2619	0.3547
$J = 0.9$	0.0505	0.0963	0.1520	0.2124

Table 1.2 Gamma Distribution: Scale parameter β_x of diseased group

2. Coverage probability

2.1 Results of Normal simulation

σ_x^2	J	Sample size (n=50, m=50)			
		NB	BCa	BP	Delta
0.5	0.4	0.9363	0.9313	0.9193	0.9350
	0.6	0.9467	0.9290	0.8776	0.9363
	0.8	0.9503	0.9310	0.8463	0.9357
	0.9	0.8813	0.9017	0.7576	0.9236
1	0.4	0.9386	0.9426	0.9147	0.9357
	0.6	0.9513	0.9117	0.8643	0.9413
	0.8	0.9410	0.9290	0.8403	0.9310
	0.9	0.8897	0.9067	0.7537	0.9333
3	0.4	0.9397	0.9400	0.9203	0.9423
	0.6	0.9477	0.9283	0.8783	0.9370
	0.8	0.9500	0.9360	0.8423	0.9333
	0.9	0.8817	0.9093	0.763	0.9266
5	0.4	0.9510	0.9510	0.9316	0.9426
	0.6	0.9543	0.9203	0.8810	0.9360
	0.8	0.9390	0.9397	0.8537	0.9293
	0.9	0.8940	0.9103	0.7590	0.9223

Table 2.1.1

σ_x^2	J	Sample size (n=50, m=80)			
		NB	BCa	BP	Delta
0.5	0.4	0.9403	0.9030	0.8977	0.9390
	0.6	0.9490	0.9040	0.8853	0.9416
	0.8	0.9577	0.9173	0.8750	0.9283
	0.9	0.9047	0.9153	0.811	0.9223
1	0.4	0.9313	0.9100	0.8983	0.9397
	0.6	0.9400	0.9057	0.8913	0.9323
	0.8	0.9457	0.9177	0.8663	0.9403
	0.9	0.9100	0.9160	0.8087	0.9233
3	0.4	0.9330	0.9210	0.9047	0.9436
	0.6	0.9487	0.9123	0.9046	0.9480
	0.8	0.9440	0.9103	0.8850	0.9327
	0.9	0.9070	0.9206	0.8227	0.9200
5	0.4	0.9376	0.9273	0.9227	0.9456
	0.6	0.9420	0.9310	0.9050	0.9403
	0.8	0.9437	0.9233	0.8767	0.9290
	0.9	0.9110	0.9293	0.8213	0.9286

Table 2.1.2

σ_x^2	J	Sample size (nd=80, nh=80)			
		NB	BCa	BP	Delta
0.5	0.4	0.9297	0.9273	0.9157	0.9473
	0.6	0.9497	0.9150	0.869	0.9446
	0.8	0.9630	0.9143	0.8473	0.9410
	0.9	0.9377	0.9143	0.7416	0.9403
1	0.4	0.9380	0.9253	0.9113	0.9453
	0.6	0.9457	0.9320	0.8880	0.9413
	0.8	0.9553	0.9160	0.8627	0.9397
	0.9	0.9407	0.9203	0.6653	0.9357
3	0.4	0.9467	0.9423	0.9253	0.9506
	0.6	0.9463	0.9207	0.8697	0.9443
	0.8	0.9623	0.9247	0.8570	0.9340
	0.9	0.9350	0.9236	0.7533	0.9223
5	0.4	0.9497	0.9400	0.9363	0.9513
	0.6	0.9570	0.9243	0.8823	0.9470
	0.8	0.9497	0.9253	0.8533	0.9420
	0.9	0.9353	0.9266	0.7733	0.9403

Table 2.1.3

σ_x^2	J	Sample size (nd=100, nh=100)			
		NB	BCa	BP	Delta
0.5	0.4	0.9436	0.9330	0.9186	0.9420
	0.6	0.9450	0.9183	0.9050	0.9400
	0.8	0.9530	0.9263	0.8783	0.9413
	0.9	0.9403	0.9353	0.8390	0.9293
1	0.4	0.9457	0.9207	0.9047	0.9427
	0.6	0.9470	0.9177	0.8943	0.9417
	0.8	0.9560	0.9200	0.8833	0.9633
	0.9	0.9440	0.9333	0.8526	0.9577
3	0.4	0.9457	0.9307	0.9243	0.9513
	0.6	0.9457	0.9230	0.9077	0.9450
	0.8	0.9563	0.9327	0.8823	0.9330
	0.9	0.9487	0.9333	0.8453	0.9373
5	0.4	0.9453	0.9427	0.9350	0.9557
	0.6	0.9570	0.9293	0.9150	0.9430
	0.8	0.9540	0.9367	0.8850	0.9423
	0.9	0.9373	0.9347	0.8497	0.9390

Table 2.1.4

σ_x^2	J	Sample size (nd=100, nh=150)			
		NB	BCa	BP	Delta
0.5	0.4	0.9263	0.9166	0.9100	0.9433
	0.6	0.9390	0.9123	0.9016	0.9460
	0.8	0.9510	0.9176	0.8853	0.9540
	0.9	0.9436	0.9147	0.8596	0.9346
1	0.4	0.9303	0.9153	0.9046	0.9433
	0.6	0.9356	0.9150	0.9033	0.9460
	0.8	0.9523	0.9166	0.8986	0.9540
	0.9	0.9527	0.9096	0.8580	0.9446
3	0.4	0.9367	0.9243	0.9210	0.9490
	0.6	0.9413	0.9186	0.9080	0.9403
	0.8	0.9520	0.9200	0.8976	0.9400
	0.9	0.9436	0.9140	0.8663	0.9423
5	0.4	0.9327	0.9293	0.9260	0.9443
	0.6	0.9466	0.9173	0.9116	0.9443
	0.8	0.9476	0.9213	0.9083	0.9400
	0.9	0.9446	0.9243	0.8840	0.9363

Table 2.1.5

σ_x^2	J	Sample size (nd=200, nh=200)			
		NB	BCa	BP	Delta
0.5	0.4	0.9380	0.9280	0.9177	0.9537
	0.6	0.9447	0.9160	0.9087	0.9446
	0.8	0.9517	0.9270	0.9093	0.9543
	0.9	0.9480	0.9293	0.8800	0.9460
1	0.4	0.9297	0.9223	0.9260	0.9467
	0.6	0.9323	0.9170	0.9020	0.9577
	0.8	0.9463	0.9290	0.9007	0.9583
	0.9	0.9550	0.9280	0.8763	0.9443
3	0.4	0.9420	0.9350	0.9263	0.9413
	0.6	0.9457	0.9210	0.9183	0.9500
	0.8	0.9593	0.9247	0.9056	0.9426
	0.9	0.9483	0.9217	0.8873	0.9366
5	0.4	0.9453	0.9373	0.9333	0.9473
	0.6	0.9430	0.9360	0.9263	0.9547
	0.8	0.9490	0.9357	0.9057	0.9446
	0.9	0.9503	0.9273	0.8893	0.9350

Table 2.1.6

2.2 Results of Gamma simulation

α_x	J	Sample size (nd=50, nh=50)			
		NB	BCa	BP	Delta
1.5	0.4	0.9503	0.9420	0.9110	0.9332
	0.6	0.9527	0.9467	0.9240	0.9245
	0.8	0.9513	0.9447	0.8956	0.9288
	0.9	0.8779	0.9347	0.8990	0.9071
2	0.4	0.9313	0.9403	0.9093	0.9436
	0.6	0.9557	0.9393	0.9103	0.9392
	0.8	0.9423	0.9533	0.9030	0.9113
	0.9	0.8783	0.9243	0.9053	0.9021
2.5	0.4	0.9420	0.9323	0.9113	0.9416
	0.6	0.9453	0.9370	0.8973	0.9376
	0.8	0.9500	0.9433	0.8996	0.9218
	0.9	0.8783	0.9226	0.8990	0.9103
3	0.4	0.9310	0.9263	0.9170	0.9432
	0.6	0.9487	0.9346	0.9100	0.9377
	0.8	0.9463	0.9456	0.8863	0.9287
	0.9	0.8747	0.9116	0.8973	0.9123

Table 2.2.1

α_x	J	Sample size (nd=50, nh=80)			
		NB	BCa	BP	Delta
1.5	0.4	0.9300	0.9090	0.9043	0.9386
	0.6	0.9460	0.9113	0.8963	0.9421
	0.8	0.9450	0.9206	0.8900	0.9466
	0.9	0.9143	0.9356	0.8850	0.9210
2	0.4	0.9310	0.8986	0.9053	0.9403
	0.6	0.9467	0.9130	0.9020	0.9487
	0.8	0.9503	0.9116	0.8843	0.9393
	0.9	0.9177	0.9426	0.8693	0.9243
2.5	0.4	0.9273	0.8960	0.8950	0.9456
	0.6	0.9500	0.8963	0.8900	0.9486
	0.8	0.9173	0.9093	0.8593	0.9330
	0.9	0.9137	0.9303	0.9063	0.9176
3	0.4	0.9333	0.9026	0.8860	0.9486
	0.6	0.9423	0.8990	0.8883	0.9396
	0.8	0.9437	0.9156	0.8786	0.9190
	0.9	0.9343	0.9380	0.8646	0.9293

Table 2.2.2

α_x	J	Sample size (nd=80, nh=80)			
		NB	BCa	BP	Delta
1.5	0.4	0.9413	0.9360	0.9193	0.9466
	0.6	0.9537	0.9410	0.9240	0.9372
	0.8	0.9487	0.9333	0.9136	0.9521
	0.9	0.9420	0.9553	0.9080	0.9456
2	0.4	0.9313	0.9400	0.9123	0.9453
	0.6	0.9430	0.9346	0.9093	0.9513
	0.8	0.9497	0.9206	0.9063	0.9533
	0.9	0.9390	0.9506	0.8996	0.9347
2.5	0.4	0.9387	0.9253	0.9126	0.9506
	0.6	0.9477	0.9390	0.9040	0.9543
	0.8	0.9563	0.9340	0.8880	0.9440
	0.9	0.9480	0.9450	0.9145	0.9247
3	0.4	0.9373	0.9256	0.9063	0.9513
	0.6	0.9397	0.9393	0.9216	0.9489
	0.8	0.9507	0.9296	0.9140	0.9427
	0.9	0.9220	0.9526	0.8903	0.9353

Table 2.2.3

α_x	J	Sample size (nd=100, nh=100)			
		NB	BCa	BP	Delta
1.5	0.4	0.9300	0.9296	0.9093	0.9487
	0.6	0.9480	0.9347	0.9287	0.9446
	0.8	0.9560	0.9476	0.9213	0.9603
	0.9	0.9363	0.9526	0.9067	0.9333
2	0.4	0.9267	0.9187	0.9173	0.9426
	0.6	0.9473	0.9307	0.9296	0.9417
	0.8	0.9617	0.9433	0.9090	0.9633
	0.9	0.9343	0.9526	0.9040	0.9577
2.5	0.4	0.9260	0.9243	0.9093	0.9457
	0.6	0.9397	0.9353	0.9187	0.9468
	0.8	0.9573	0.9367	0.9150	0.9336
	0.9	0.9457	0.9506	0.8997	0.9277
3	0.4	0.9377	0.9273	0.9106	0.9513
	0.6	0.9420	0.9290	0.9167	0.9521
	0.8	0.9553	0.9436	0.8983	0.9467
	0.9	0.9370	0.9527	0.8933	0.9363

Table 2.2.4

α_x	J	Sample size (nd=100, nh=150)			
		NB	BCa	BP	Delta
1.5	0.4	0.9303	0.9133	0.9166	0.9449
	0.6	0.9396	0.9143	0.9120	0.9503
	0.8	0.9513	0.9223	0.9080	0.9487
	0.9	0.9487	0.9286	0.9003	0.9377
2	0.4	0.9340	0.9093	0.9123	0.9396
	0.6	0.9447	0.9126	0.9120	0.9530
	0.8	0.9500	0.9173	0.9066	0.9450
	0.9	0.9463	0.9236	0.8946	0.9456
2.5	0.4	0.9306	0.9140	0.9023	0.9474
	0.6	0.9486	0.9250	0.9130	0.9531
	0.8	0.9566	0.9146	0.8960	0.9447
	0.9	0.9453	0.9193	0.8830	0.9363
3	0.4	0.9247	0.9050	0.9033	0.9497
	0.6	0.9423	0.9103	0.9170	0.9493
	0.8	0.9513	0.9170	0.8950	0.9480
	0.9	0.9467	0.9246	0.8836	0.9367

Table 2.2.5

α_x	J	Sample size (nd=200, nh=200)			
		NB	BCa	BP	Delta
1.5	0.4	0.9383	0.9293	0.9163	0.9543
	0.6	0.9400	0.9336	0.9193	0.9423
	0.8	0.9537	0.9353	0.9233	0.9520
	0.9	0.9510	0.9463	0.9220	0.9423
2	0.4	0.9250	0.9337	0.9190	0.9456
	0.6	0.9363	0.9300	0.9283	0.9577
	0.8	0.9460	0.9260	0.9217	0.9483
	0.9	0.9500	0.9467	0.9177	0.9436
2.5	0.4	0.9293	0.9227	0.9173	0.9467
	0.6	0.9360	0.9260	0.9093	0.9513
	0.8	0.9513	0.9320	0.9223	0.9430
	0.9	0.9520	0.9387	0.9080	0.9393
3	0.4	0.9373	0.9163	0.9167	0.9497
	0.6	0.9453	0.9393	0.9170	0.9526
	0.8	0.9503	0.9327	0.9097	0.9456
	0.9	0.9470	0.9457	0.9100	0.9376

Table 2.2.6

3. Average interval length

3.1 Results from normal distribution

σ_x^2	J	Sample size (nd=50, nh=50)			
		NB	BCa	BP	Delta
0.5	0.4	0.3036	0.3243	0.3254	0.2796
	0.6	0.2660	0.2889	0.2859	0.2496
	0.8	0.1942	0.2138	0.2077	0.1839
	0.9	0.1315	0.1523	0.1390	0.1230
1	0.4	0.3105	0.3321	0.3328	0.2802
	0.6	0.2696	0.2902	0.2878	0.2475
	0.8	0.1946	0.2147	0.2072	0.1843
	0.9	0.1297	0.1525	0.1378	0.1224
3	0.4	0.3019	0.3233	0.3239	0.2696
	0.6	0.2678	0.2888	0.2869	0.2453
	0.8	0.1947	0.2157	0.2075	0.1817
	0.9	0.1317	0.1531	0.1394	0.1234
5	0.4	0.2950	0.3147	0.3160	0.2568
	0.6	0.2656	0.2858	0.2838	0.2390
	0.8	0.1953	0.2166	0.2082	0.1805
	0.9	0.1325	0.1553	0.1389	0.1241

Table 3.1.1

σ_x^2	J	Sample size (nd=50, nh=80)			
		NB	BCa	BP	Delta
0.5	0.4	0.2739	0.2882	0.2886	0.2463
	0.6	0.2392	0.2537	0.2535	0.2193
	0.8	0.1744	0.1878	0.1844	0.1604
	0.9	0.1188	0.1319	0.1243	0.1078
1	0.4	0.2853	0.3014	0.3022	0.2498
	0.6	0.2475	0.2620	0.2627	0.2254
	0.8	0.1777	0.1933	0.1893	0.1623
	0.9	0.1214	0.1364	0.1274	0.1139
3	0.4	0.2844	0.3031	0.3044	0.2521
	0.6	0.2525	0.2700	0.2699	0.2298
	0.8	0.1851	0.1995	0.1981	0.1717
	0.9	0.1255	0.1411	0.1330	0.1159
5	0.4	0.2792	0.2981	0.2989	0.2437
	0.6	0.2528	0.2718	0.2709	0.2276
	0.8	0.1867	0.2041	0.1996	0.1729
	0.9	0.1286	0.1458	0.1353	0.1187

Table 3.1.2

σ_x^2	J	Sample size (nd=80, nh=80)			
		NB	BCa	BP	Delta
0.5	0.4	0.2507	0.2610	0.3242	0.2216
	0.6	0.2210	0.2310	0.2856	0.1982
	0.8	0.1626	0.1726	0.2077	0.1452
	0.9	0.1137	0.1232	0.1369	0.0980
1	0.4	0.2566	0.2677	0.2685	0.2187
	0.6	0.2231	0.2336	0.2331	0.1976
	0.8	0.1623	0.1727	0.1699	0.1453
	0.9	0.1118	0.1231	0.1700	0.0989
3	0.4	0.2485	0.2583	0.3253	0.2138
	0.6	0.2209	0.2307	0.2870	0.1945
	0.8	0.1630	0.1728	0.2083	0.1445
	0.9	0.1132	0.1238	0.1383	0.0973
5	0.4	0.2403	0.2507	0.3154	0.2034
	0.6	0.2183	0.2286	0.2842	0.1896
	0.8	0.1633	0.1735	0.2082	0.1431
	0.9	0.1136	0.1246	0.1404	0.0984

Table 3.1.3

σ_x^2	J	Sample size (nd=100, nh=100)			
		NB	BCa	BP	Delta
0.5	0.4	0.2276	0.2341	0.2361	0.1983
	0.6	0.2011	0.2074	0.2091	0.1772
	0.8	0.1487	0.1553	0.1533	0.1303
	0.9	0.1037	0.1110	0.1065	0.0874
1	0.4	0.2330	0.2394	0.2415	0.1934
	0.6	0.2025	0.2098	0.2100	0.1756
	0.8	0.1486	0.1553	0.1540	0.1289
	0.9	0.1031	0.1106	0.1068	0.0876
3	0.4	0.2251	0.2309	0.2330	0.1913
	0.6	0.2008	0.2069	0.2086	0.1743
	0.8	0.1491	0.1549	0.1538	0.1292
	0.9	0.1041	0.1115	0.1073	0.0874
5	0.4	0.2178	0.2233	0.2260	0.1820
	0.6	0.1981	0.2049	0.2059	0.1696
	0.8	0.1489	0.1563	0.1533	0.1282
	0.9	0.1049	0.1118	0.1069	0.0873

Table 3.1.4

σ_x^2	J	Sample size (nd=100, nh=150)			
		NB	BCa	BP	Delta
0.5	0.4	0.2064	0.2097	0.2114	0.1776
	0.6	0.1818	0.1853	0.1869	0.1584
	0.8	0.1344	0.1383	0.1381	0.1159
	0.9	0.0946	0.0980	0.0963	0.0774
1	0.4	0.2154	0.2189	0.2217	0.1741
	0.6	0.1871	0.1914	0.1926	0.1617
	0.8	0.1379	0.1412	0.1414	0.1197
	0.9	0.0960	0.1002	0.0980	0.0867
3	0.4	0.2118	0.2177	0.2206	0.1802
	0.6	0.1901	0.1946	0.1967	0.1643
	0.8	0.1413	0.1460	0.1459	0.1227
	0.9	0.0988	0.1041	0.1018	0.0830
5	0.4	0.2074	0.2129	0.2158	0.1737
	0.6	0.1896	0.1952	0.1964	0.1623
	0.8	0.1425	0.1475	0.1476	0.1230
	0.9	0.1002	0.1056	0.1041	0.0845

Table 3.1.5

σ_x^2	J	Sample size (nd=200, nh=200)			
		NB	BCa	BP	Delta
0.5	0.4	0.1668	0.1678	0.1696	0.1405
	0.6	0.1474	0.1491	0.1502	0.1257
	0.8	0.1100	0.1118	0.1118	0.0923
	0.9	0.0780	0.0801	0.0791	0.0620
1	0.4	0.1710	0.1724	0.1742	0.1376
	0.6	0.1488	0.1504	0.1515	0.1242
	0.8	0.1103	0.1119	0.1119	0.0920
	0.9	0.0778	0.0803	0.0794	0.0627
3	0.4	0.1637	0.1650	0.1668	0.1354
	0.6	0.1467	0.1483	0.1493	0.1235
	0.8	0.1100	0.1112	0.1119	0.0919
	0.9	0.0783	0.0800	0.0795	0.0619
5	0.4	0.1574	0.1587	0.1605	0.1289
	0.6	0.1444	0.1460	0.1477	0.1201
	0.8	0.1091	0.1111	0.1116	0.0910
	0.9	0.0786	0.0801	0.0795	0.0623

Table 3.1.6

3.2 Results from Gamma distribution

α_x	J	Sample size (nd=50, nh=50)			
		NB	BCa	BP	Delta
1.5	0.4	0.3036	0.3243	0.3254	0.2841
	0.6	0.2660	0.2889	0.2859	0.2502
	0.8	0.1942	0.2138	0.2077	0.1821
	0.9	0.1315	0.1523	0.1390	0.1245
2	0.4	0.3105	0.3321	0.3328	0.2833
	0.6	0.2696	0.2902	0.2878	0.2442
	0.8	0.1946	0.2147	0.2072	0.1824
	0.9	0.1297	0.1525	0.1378	0.1236
2.5	0.4	0.3019	0.3233	0.3239	0.2732
	0.6	0.2678	0.2888	0.2869	0.2439
	0.8	0.1947	0.2157	0.2075	0.1803
	0.9	0.1317	0.1531	0.1394	0.1244
3	0.4	0.2950	0.3147	0.3160	0.2668
	0.6	0.2656	0.2858	0.2838	0.2411
	0.8	0.1953	0.2166	0.2082	0.1813
	0.9	0.1325	0.1553	0.1389	0.1277

Table 3.2.1

α_x	J	Sample size (nd=50, nh=80)			
		NB	BCa	BP	Delta
1.5	0.4	0.3036	0.3243	0.3254	0.2737
	0.6	0.2660	0.2889	0.2859	0.2397
	0.8	0.1942	0.2138	0.2077	0.1824
	0.9	0.1315	0.1523	0.1390	0.1268
2	0.4	0.3105	0.3321	0.3328	0.2738
	0.6	0.2696	0.2902	0.2878	0.2474
	0.8	0.1946	0.2147	0.2072	0.1833
	0.9	0.1297	0.1525	0.1378	0.1145
2.5	0.4	0.3019	0.3233	0.3239	0.2713
	0.6	0.2678	0.2888	0.2869	0.2328
	0.8	0.1947	0.2157	0.2075	0.1747
	0.9	0.1317	0.1531	0.1394	0.1239
3	0.4	0.2950	0.3147	0.3160	0.2647
	0.6	0.2656	0.2858	0.2838	0.2386
	0.8	0.1953	0.2166	0.2082	0.1819
	0.9	0.1325	0.1553	0.1389	0.1243

Table 3.2.2

α_x	J	Sample size (nd=80, nh=80)			
		NB	BCa	BP	Delta
1.5	0.4	0.3036	0.3243	0.3254	0.2726
	0.6	0.2660	0.2889	0.2859	0.2352
	0.8	0.1942	0.2138	0.2077	0.1732
	0.9	0.1315	0.1523	0.1390	0.1284
2	0.4	0.3105	0.3321	0.3328	0.2767
	0.6	0.2696	0.2902	0.2878	0.2335
	0.8	0.1946	0.2147	0.2072	0.1763
	0.9	0.1297	0.1525	0.1378	0.1079
2.5	0.4	0.3019	0.3233	0.3239	0.2658
	0.6	0.2678	0.2888	0.2869	0.2425
	0.8	0.1947	0.2157	0.2075	0.1815
	0.9	0.1317	0.1531	0.1394	0.1098
3	0.4	0.2950	0.3147	0.3160	0.2574
	0.6	0.2656	0.2858	0.2838	0.2436
	0.8	0.1953	0.2166	0.2082	0.1781
	0.9	0.1325	0.1553	0.1389	0.1164

Table 3.2.3

α_x	J	Sample size (nd=100, nh=100)			
		NB	BCa	BP	Delta
1.5	0.4	0.2308	0.2369	0.2390	0.2073
	0.6	0.2005	0.2073	0.2088	0.1746
	0.8	0.1488	0.1550	0.1538	0.1327
	0.9	0.1045	0.1117	0.1078	0.0914
2	0.4	0.2329	0.2392	0.2413	0.1987
	0.6	0.2026	0.2087	0.2095	0.1749
	0.8	0.1485	0.1551	0.1532	0.1294
	0.9	0.1043	0.1116	0.1070	0.0901
2.5	0.4	0.2329	0.2391	0.2415	0.2006
	0.6	0.2030	0.2092	0.2100	0.1788
	0.8	0.1483	0.1551	0.1533	0.1268
	0.9	0.1042	0.1118	0.1066	0.0898
3	0.4	0.2318	0.2382	0.2399	0.2023
	0.6	0.2029	0.2092	0.2100	0.1776
	0.8	0.1485	0.1551	0.1531	0.1274
	0.9	0.1035	0.1104	0.1061	0.0889

Table 3.2.4

α_x	J	Sample size (nd=100, nh=150)			
		NB	BCa	BP	Delta
1.5	0.4	0.2155	0.2208	0.2229	0.1886
	0.6	0.1897	0.1945	0.1963	0.1596
	0.8	0.1420	0.1464	0.1462	0.1239
	0.9	0.1009	0.1059	0.1030	0.0884
2	0.4	0.2163	0.2206	0.2227	0.1769
	0.6	0.1890	0.1940	0.1952	0.1623
	0.8	0.1410	0.1454	0.1455	0.1285
	0.9	0.1000	0.1047	0.1026	0.0947
2.5	0.4	0.2147	0.2188	0.2209	0.1856
	0.6	0.1882	0.1931	0.1939	0.1613
	0.8	0.1401	0.1444	0.1437	0.1258
	0.9	0.0990	0.1037	0.1012	0.0868
3	0.4	0.2123	0.2162	0.2179	0.1857
	0.6	0.1869	0.1910	0.1933	0.1603
	0.8	0.1392	0.1434	0.1423	0.1198
	0.9	0.0980	0.1028	0.1005	0.0844

Table 3.2.5

α_x	J	Sample size (nd=200, nh=200)			
		NB	BCa	BP	Delta
1.5	0.4	0.1685	0.1699	0.1714	0.1415
	0.6	0.1473	0.1486	0.1497	0.1277
	0.8	0.1095	0.1114	0.1116	0.0903
	0.9	0.0782	0.0804	0.0794	0.0680
2	0.4	0.1706	0.1720	0.1737	0.1376
	0.6	0.1481	0.1497	0.1509	0.1242
	0.8	0.1097	0.1115	0.1118	0.0920
	0.9	0.0780	0.0797	0.0792	0.0627
2.5	0.4	0.1706	0.1722	0.1734	0.1414
	0.6	0.1487	0.1506	0.1515	0.1275
	0.8	0.1102	0.1116	0.1119	0.1019
	0.9	0.0778	0.0800	0.0792	0.0713
3	0.4	0.1696	0.1711	0.1728	0.1379
	0.6	0.1491	0.1504	0.1511	0.1231
	0.8	0.1097	0.1118	0.1117	0.0974
	0.9	0.0778	0.0800	0.0789	0.0695

Table 3.2.6

APPENDIX B: S-Plus Code for Simulation

```
###code for NB method, normal case###
alpha<-0.05
k<-qnorm(1-alpha/2)

nd<-200
nh<-200
m<-3000

uh<-0
sh<-sqrt(1)

#ud<-c(0.8484,1.4071,2.1682,2.7927)
#sd<-sqrt(0.5)

#ud<-c(1.0489,1.6833,2.5632,3.2898)
#sd<-sqrt(1)

#ud<-c(1.2540,2.1843,3.4247,4.4340)
#sd<-sqrt(3)

#ud<-c(1.2815,2.4493,3.9625,5.1777)
#sd<-sqrt(5)

#Rtt<-c(1.400017,1.599987,1.799994,1.899996)

for (q in 1:4)
{

j<-function(xd, xh, b, k)
{
   r<-rep(NA, b)
   if (b>1)
      {
         for (i in 1:b)
         {
            t<-sample(xd, length(xd), replace=T)
            u<-sample(xh, length(xh), replace=T)
            if (max(t)<min(u))
            {
               r[i]<-0
            }
            else
            {
               r[i]<-
   (sum(t>=chat)+k^2/2)/(length(t)+k^2)+(sum(u<=chat)+k^2/2)/(length(u)+k^2)
            }
         }
      }
      else r[1]<-sum(xd>=chat)/length(xd)+sum(xh<=chat)/length(xh)
      return(r)
}

mnb<-0
mnbr<-0
LUb<-0


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

```
    xh<-rnorm(nh,uh,sh)
    xd<-rnorm(nd,ud[q],sd)

    z<-c(xd,xh)
    z<-sort(z)
    c<-z
    result<-rep(0,nd+nh)
for (i in 1:(nd+nh))
{
    if (max(xd)<min(xh))
    {
       result<-0
    }
    else
    {
       result[i]<-
    (sum(xd>=c[i])+k^2/2)/(length(xd)+k^2)+(sum(xh<=c[i])+k^2/2)/(length(xh)+k^2
    )
   }

}
result

mm<-max(result)
mm

l<-1
for (i in 1:(nh+nd))
{
    if (result[l]!=mm)
    l<-l+1
    else l<-l
}
l

chat<-c[l]

    test<-j(xd,xh,300,k)
    bmu<-mean(test)
    bstd<-sqrt(var(test))
    mnb[i]<-((bmu-k*bstd)<=Rtt[q])*(Rtt[q]<=(bmu+k*bstd))
    mnbr[i]<-(Rtt[q]>(bmu+k*bstd))
    LUb[i]<-2*k*bstd
}
mnb<-sort(mnb)
cov.bn<-mean(mnb)
cov.bnerr<-mean(mnbr)
LUb<-sort(LUb)
wid.bn<-mean(LUb)
sink("RocBootstrapoutBNN-4689-0.5-2020", append=T)

cat("the coverage of AC adjust bootstrap CI for R(t):", cov.bn, "\n")
cat("the width of AC adjust bootstrap(BII) CI for R(t):", wid.bn, "\n")
cat("the lower coverage err of AC adjust bootstrap(BII) CI for R(t):", 1-
    cov.bn-cov.bnerr, "\n")
cat("the upper coverage err of AC adjust bootstrap(BII) CI for R(t):",
    cov.bnerr, "\n")
sink()
}

###code for NB method, gamma case###

alpha<-0.05
```

```
k<-qnorm(1-alpha/2)
k

nd<-80
nh<-80
m<-3000

ah<-1.5
bh<-1

#ad<-1.5
#bd<-c(0.4027665,0.2295403,0.102192,0.0504986)

#ad<-2
#bd<-c(0.6016187,0.3616621,0.1769386,0.09630156)

#ad<-2.5
#bd<-c(0.8188774,0.5064433,0.2619157,0.1509695)

ad<-3
bd<-c(1.052237,0.6614291,0.3547253,0.2124167)

Rtt<-c(1.4,1.6,1.8,1.9)

for (q in 1:4)
{
j<-function(xd, xh, b, k)
{
   r<-rep(NA, b)
   if (b>1)
      {
         for (i in 1:b)
         {
            t<-sample(xd, length(xd), replace=T)
            u<-sample(xh, length(xh), replace=T)
            if (max(t)<min(u))
            {
               r[i]<-0
            }
            else
            {
               r[i]<-
   (sum(t>=chat)+k^2/2)/(length(t)+k^2)+(sum(u<=chat)+k^2/2)/(length(u)+k^2)
            }
         }
      }
      else r[1]<-sum(xd>=chat)/length(xd)+sum(xh<=chat)/length(xh)
      return(r)
}

mnb<-0
mnbr<-0
LUb<-0


for (i in 1:m)
{
   xh<-rgamma(nh,ah,bh)
   xd<-rgamma(nd,ad,bd[q])

   z<-c(xd,xh)
   z<-sort(z)
   c<-z
```

```
   result<-rep(0,nd+nh)
for (i in 1:(nd+nh))
{
   if (max(xd)< min(xh))
   {
      result<-0
   }
   else
   {
      result[i]<-
   (sum(xd>=c[i])+k^2/2)/(length(xd)+k^2)+(sum(xh<=c[i])+k^2/2)/(length(xh)+k^2
   )
   }

}
result

mm<-max(result)
mm

l<-1
for (i in 1:(nh+nd))
{
   if (result[l]!=mm)
   l<-l+1
   else l<-l
}
l

chat<-c[l]

   test<-j(xd,xh,300,k)
   bmu<-mean(test)
   bstd<-sqrt(var(test))
   mnb[i]<-((bmu-k*bstd)<=Rtt[q])*(Rtt[q]<=(bmu+k*bstd))
   mnbr[i]<-(Rtt[q]>(bmu+k*bstd))
   LUb[i]<-2*k*bstd
}
mnb<-sort(mnb)
cov.bn<-mean(mnb)
cov.bnerr<-mean(mnbr)
LUb<-sort(LUb)
wid.bn<-mean(LUb)

sink("RocBootstrapoutBIIGamma-4689-3-8080", append=T)

cat("the coverage of AC adjust bootstrap CI for R(t):", cov.bn, "\n")
cat("the width of AC adjust bootstrap(BII) CI for R(t):", wid.bn, "\n")
cat("the lower coverage err of AC adjust bootstrap(BII) CI for R(t):", 1-
   cov.bn-cov.bnerr, "\n")
cat("the upper coverage err of AC adjust bootstrap(BII) CI for R(t):",
   cov.bnerr, "\n")
sink()
}

### code for BCa method, normal case###
nd<-100
nh<-150
w<-3000

uh<-0
sh<-sqrt(1)
```

```
#ud<-c(0.8484,1.4071,2.1682,2.7927)
#sd<-sqrt(0.5)

#ud<-c(1.0489,1.6833,2.5632,3.2898)
#sd<-sqrt(1)

#ud<-c(1.2540,2.1843,3.4247,4.4340)
#sd<-sqrt(3)

ud<-c(1.2815,2.4493,3.9625,5.1777)
sd<-sqrt(5)

Rtt<-c(1.400017,1.599987,1.799994,1.899996)


for (k in 1:4)
{
jackj<-function(xd, xh, group.size=1)
{
   g<-group.size
   m<-floor(length(xd)/g)+floor(length(xh)/g)
   rr<-rep(NA, m)
   for (i in 1:floor(length(xd)/g))
   {
      t<-xd[-c(((i-1)*g+1):(i*g))]
      u<-xh
      rr[i]<-(sum(t>=chat))/(length(t))+(sum(u<=chat))/(length(u))
   }
   for (i in 1:floor(length(xh)/g))
   {
      t<-xd
      u<-xh[-c(((i-1)*g+1):(i*g))]
      rr[floor(length(xd)/g)+i]<-
   (sum(t>=chat))/(length(t))+(sum(u<=chat))/(length(u))
   }
   return(rr)
}

youd<-function(xd, xh, b)
{
   r<-rep(NA, b)
   if (b>1)
      {
         for (i in 1:b)
         {
            t<-sample(xd, length(xd), replace=T)
            u<-sample(xh, length(xh), replace=T)
            if (max(t)<min(u))
            {
               r[i]<-0
            }
            else
            {
               r[i]<-sum(t>=chat)/length(t)+sum(u<=chat)/length(u)
            }
         }
      }
      #else r[1]<-sum(xd>=chat)/length(xd)+sum(xh<=chat)/length(xh)
      return(r)
}
```

```
my.limits.bca<-function(xd, xh, obs, boot.reps, probs=c(25,50,950,975)/1000,
   details=F,
z0=NULL, acceleration=NULL, group.size=NULL)
{
   if(missing(group.size)) group.size<-max(1,
   floor(length(xd)/10)+floor(length(xh)/10))
      boot.reps<-as.matrix(boot.reps)
      accel<-acceleration
      if(is.null(accel))
      {
         accel<-0
         jackvalues<-jackj(xd,xh,group.size)
         thetahat<-mean(jackvalues)
         accel<-sum((thetahat-jackvalues)^3)/(6*(sum((thetahat-
   jackvalues)^2)^(3/2)))
      }
      zprobs<-qnorm(probs)
      if(is.null(z0))
      {
         z0<-rep(0,length(obs))
         for (i in 1:length(obs))
            z0[i]<-qnorm(mean(boot.reps[, i]<obs[i], na.rm=T))
      }
      names(z0)<-names(probs)
      emp.probs<-bca.percent<-matrix(nrow=length(obs), ncol=length(probs))
      for (i in 1:length(obs))
      {
         if(z0[i]==-Inf)
         {
            emp.probs[i, ]<-rep(0, length(emp.probs[i, ]))
            bca.percent[i, ]<-rep(min(boot.reps[, i]), length(bca.percent[i,
   ]))
         }
         else if(z0==Inf||is.na(accel))
         {
            emp.probs[i, ]<-rep(1, length(emp.probs[i, ]))
            bca.percent[i, ]<-rep(max(boot.reps[, i]), length(bca.percent[i,
   ]))
         }
         else
         {
            emp.probs[i, ]<-pnorm(z0[i]+(z0[i]+zprobs)/(1-
   accel[i]*(z0[i]+zprobs)))
            bca.percent[i, ]<-quantile(boot.reps[, i], emp.probs[i, ],
   na.rm=T)
         }
      }
      dimnames(bca.percent)<-list(names(obs), paste(100*probs, "%", sep=""))
      if(details)
      {
         dimnames(emp.probs)<-dimnames(bca.percent)
         bca.percent<-list(limits=bca.percent, emp.probs=emp.probs, z0=z0,
   acceleration=accel
         , group.size=group.size)
      }
      bca.percent
}


mn3<-0
mn3r<-0
LU3<-0
```

```
for (i in 1:w)
{
   xh<-rnorm(nh,uh,sh)
   xd<-rnorm(nd,ud[k],sd)

   h<-c(xd,xh)
   h<-sort(h)
   c<-h
   result<-rep(0,nd+nh)
for (i in 1:(nd+nh))
{
   if (max(xd)<min(xh))
   {
      result<-0
   }
   else
   {
      result[i]<-sum(xd>=c[i])/length(xd)+sum(xh<=c[i])/length(xh)
    }
}
result

mm<-max(result)
mm

l<-1
for (i in 1:(nh+nd))
{
   if (result[l]!=mm)
   l<-l+1
   else l<-l

}
l
chat<-c[l]

youd1<-sum(xd>=chat)/length(xd)+sum(xh<=chat)/length(xh)
youd1


test<-youd(xd,xh,300)
btci<-my.limits.bca(xd,xh,youd1,test,probs=c(25,50,950,975)/1000)
Lbt<-btci[1]
Ubt<-btci[4]

mn3[i]<-(Lbt<=Rtt[k])*(Rtt[k]<=Ubt)
mn3r[i]<-(Rtt[k]>Ubt)
LU3[i]<-Ubt-Lbt
}

mn3<-sort(mn3)
cov.bt<-mean(mn3)
cov.bterr<-mean(mn3r)
LU3<-sort(LU3)
wid.bt<-mean(LU3)

sink("RocBootstrapouttryBCaN-4689-5-1015", append=T)

cat("the coverage of BCa bootstrap CI for R(t):", cov.bt, "\n")
cat("the width of AC adjust bootstrap(BII) CI for R(t):", wid.bt, "\n")
cat("the lower coverage err of AC adjust bootstrap(BII) CI for R(t):", 1-
   cov.bt-cov.bterr, "\n")
```

```
cat("the upper coverage err of AC adjust bootstrap(BII) CI for R(t):",
   cov.bterr, "\n")
sink()
}

### code for BCa method, gamma case###
nd<-80
nh<-80
w<-3000

ah<-1.5
bh<-1

#ad<-1.5
#bd<-c(0.4027665, 0.2295403, 0.102192, 0.0504986)

#ad<-2
#bd<-c(0.6016187, 0.3616621, 0.1769386, 0.09630156)

#ad<-2.5
#bd<-c(0.8188774, 0.5064433, 0.2619157, 0.1509695)

#ad<-3
#bd<-c(1.052237, 0.6614291, 0.3547253, 0.2124167)

Rtt<-c(1.4, 1.6, 1.8, 1.9)

for (k in 1:4)
{


jackj<-function(xd, xh, group.size=1)
{
   g<-group.size
   m<-floor(length(xd)/g)+floor(length(xh)/g)
   rr<-rep(NA, m)
   for (i in 1:floor(length(xd)/g))
   {
      t<-xd[-c(((i-1)*g+1):(i*g))]
      u<-xh
      rr[i]<-(sum(t>=chat))/(length(t))+(sum(u<=chat))/(length(u))
   }
   for (i in 1:floor(length(xh)/g))
   {
      t<-xd
      u<-xh[-c(((i-1)*g+1):(i*g))]
      rr[floor(length(xd)/g)+i]<-
   (sum(t>=chat))/(length(t))+(sum(u<=chat))/(length(u))
   }
   return(rr)
}

youd<-function(xd, xh, b)
{
   r<-rep(NA, b)
   if (b>1)
      {
         for (i in 1:b)
         {
            t<-sample(xd, length(xd), replace=T)
            u<-sample(xh, length(xh), replace=T)
            if (max(t)<min(u))
            {
```

```
                r[i]<-0
            }
            else
            {
                r[i]<-sum(t>=chat)/length(t)+sum(u<=chat)/length(u)
            }
        }
    }
    #else r[1]<-sum(xd>=chat)/length(xd)+sum(xh<=chat)/length(xh)
    return(r)
}


my.limits.bca<-function(xd, xh, obs, boot.reps, probs=c(25,50,950,975)/1000,
   details=F,
z0=NULL, acceleration=NULL, group.size=NULL)
{
   if(missing(group.size)) group.size<-max(1,
   floor(length(xd)/10)+floor(length(xh)/10))
      boot.reps<-as.matrix(boot.reps)
      accel<-acceleration
      if(is.null(accel))
      {
         accel<-0
         jackvalues<-jackj(xd,xh,group.size)
         thetahat<-mean(jackvalues)
         accel<-sum((thetahat-jackvalues)^3)/(6*(sum((thetahat-
   jackvalues)^2)^(3/2)))
      }
      zprobs<-qnorm(probs)
      if(is.null(z0))
      {
         z0<-rep(0,length(obs))
         for (i in 1:length(obs))
            z0[i]<-qnorm(mean(boot.reps[, i]<obs[i], na.rm=T))
      }
      names(z0)<-names(probs)
      emp.probs<-bca.percent<-matrix(nrow=length(obs), ncol=length(probs))
      for (i in 1:length(obs))
      {
         if(z0[i]==-Inf)
         {
            emp.probs[i, ]<-rep(0, length(emp.probs[i, ]))
            bca.percent[i, ]<-rep(min(boot.reps[, i]), length(bca.percent[i,
   ]))
         }
         else if(z0==Inf||is.na(accel))
         {
            emp.probs[i, ]<-rep(1, length(emp.probs[i, ]))
            bca.percent[i, ]<-rep(max(boot.reps[, i]), length(bca.percent[i,
   ]))
         }
         else
         {
            emp.probs[i, ]<-pnorm(z0[i]+(z0[i]+zprobs)/(1-
   accel[i]*(z0[i]+zprobs)))
            bca.percent[i, ]<-quantile(boot.reps[, i], emp.probs[i, ],
   na.rm=T)
         }
      }
      dimnames(bca.percent)<-list(names(obs), paste(100*probs, "%", sep=""))
      if(details)
      {
```

```
          dimnames(emp.probs)<-dimnames(bca.percent)
          bca.percent<-list(limits=bca.percent, emp.probs=emp.probs, z0=z0,
   acceleration=accel
          , group.size=group.size)
       }
       bca.percent
}


mn3<-0
mn3r<-0
LU3<-0



for (i in 1:w)
{
   xh<-rgamma(nh,ah,bh)
   xd<-rgamma(nd,ad,bd[k])

   h<-c(xd,xh)
   h<-sort(h)
   c<-h
   result<-rep(0,nd+nh)
for (i in 1:(nd+nh))
{
   if (max(xd)<min(xh))
   {
      result<-0
   }
   else
   {
      result[i]<-sum(xd>=c[i])/length(xd)+sum(xh<=c[i])/length(xh)
    }
}
result

mm<-max(result)
mm

l<-1
for (i in 1:(nh+nd))
{
   if (result[l]!=mm)
   l<-l+1
   else l<-l

}
l
chat<-c[l]

youd1<-sum(xd>=chat)/length(xd)+sum(xh<=chat)/length(xh)
youd1


test<-youd(xd,xh,300)
btci<-my.limits.bca(xd,xh,youd1,test,probs=c(25,50,950,975)/1000)
Lbt<-btci[1]
Ubt<-btci[4]

mn3[i]<-(Lbt<=Rtt[k])*(Rtt[k]<=Ubt)
mn3r[i]<-(Rtt[k]>Ubt)
LU3[i]<-Ubt-Lbt
```

```
}

mn3<-sort(mn3)
cov.bt<-mean(mn3)
cov.bterr<-mean(mn3r)
LU3<-sort(LU3)
wid.bt<-mean(LU3)


sink("RocBootstrapouttryBCaGamma-4689-3-8080", append=T)

cat("the coverage of BCa bootstrap CI for R(t):", cov.bt, "\n")
cat("the width of AC adjust bootstrap(BII) CI for R(t):", wid.bt, "\n")
cat("the lower coverage err of AC adjust bootstrap(BII) CI for R(t):", 1-
   cov.bt-cov.bterr, "\n")
cat("the upper coverage err of AC adjust bootstrap(BII) CI for R(t):",
   cov.bterr, "\n")

sink()
}

### code for BP method, normal case###
nd<-100
nh<-150
w<-3000

uh<-0
sh<-sqrt(1)

#ud<-c(0.8484,1.4071,2.1682,2.7927)
#sd<-sqrt(0.5)

#ud<-c(1.0489,1.6833,2.5632,3.2898)
#sd<-sqrt(1)

#ud<-c(1.2540,2.1843,3.4247,4.4340)
#sd<-sqrt(3)

ud<-c(1.2815,2.4493,3.9625,5.1777)
sd<-sqrt(5)

Rtt<-c(1.400017,1.599987,1.799994,1.899996)

for (q in 1:4)
{
j<-function(xd, xh, b)
{
   r<-rep(NA, b)
   if (b>1)
      {
         for (i in 1:b)
         {
            t<-sample(xd, length(xd), replace=T)
            u<-sample(xh, length(xh), replace=T)
            if (max(t)<min(u))
            {
               r[i]<-0
            }
            else
            {
               r[i]<-sum(t>=chat)/length(t)+sum(u<=chat)/length(u)
            }
         }
```

```
       }
       return(r)
}

mnb<-0
mnbr<-0
Lb<-0
Ub<-0
LUb<-0


for (i in 1:w)
{
   xh<-rnorm(nh,uh,sh)
   xd<-rnorm(nd,ud[q],sd)

   z<-c(xd,xh)
   z<-sort(z)
   c<-z
   result<-rep(0,nd+nh)
for (i in 1:(nd+nh))
{
   if (max(xd)<min(xh))
   {
      result<-0
   }
   else
   {
      result[i]<-sum(xd>=c[i])/length(xd)+sum(xh<=c[i])/length(xh)
   }
}
result

mm<-max(result)
mm

l<-1
for (i in 1:(nh+nd))
{
   if (result[l]!=mm)
   l<-l+1
   else l<-l
}
l

chat<-c[l]

   test<-j(xd,xh,300)
   tt<-sort(test)
   bmu<-mean(tt)
   #bstd<-sqrt(var(test))
   Lb[i]<-(tt[7]+tt[8])/2
   Ub[i]<-(tt[293]+tt[292])/2
   mnb[i]<-(Lb[i]<=Rtt[q])*(Rtt[q]<=Ub[i])
   mnbr[i]<-(Rtt[q]>Ub[i])
   LUb[i]<-Ub[i]-Lb[i]
}
mnb<-sort(mnb)
cov.bn<-mean(mnb)
cov.bnerr<-mean(mnbr)
LUb<-sort(LUb)
wid.bn<-mean(LUb)
```

```
sink("RocBootstrapouttryBPN-4689-5-1015", append=T)

cat("the coverage of BP bootstrap CI for R(t):", cov.bn, "\n")
cat("the width of AC adjust bootstrap(BII) CI for R(t):", wid.bn, "\n")
cat("the lower coverage err of AC adjust bootstrap(BII) CI for R(t):", 1-
   cov.bn-cov.bnerr, "\n")
cat("the upper coverage err of AC adjust bootstrap(BII) CI for R(t):",
   cov.bnerr, "\n")
sink()
}

### code for BP method, gamma case###
nd<-80
nh<-80
m<-3000

ah<-1.5
bh<-1

#ad<-1.5
#bd<-c(0.4027665, 0.2295403, 0.102192, 0.0504986)

#ad<-2
#bd<-c(0.6016187, 0.3616621, 0.1769386, 0.09630156)

#ad<-2.5
#bd<-c(0.8188774, 0.5064433, 0.2619157, 0.1509695)

ad<-3
bd<-c(1.052237, 0.6614291, 0.3547253, 0.2124167)

Rtt<-c(1.4, 1.6, 1.8, 1.9)

for (p in 1:4)
{


j<-function(xd, xh, b)
{
   r<-rep(NA, b)
   if (b>1)
      {
         for (i in 1:b)
         {
            t<-sample(xd, length(xd), replace=T)
            u<-sample(xh, length(xh), replace=T)
            if (max(t)<min(u))
            {
               r[i]<-0
            }
            else
            {
               r[i]<-sum(t>=chat)/length(t)+sum(u<=chat)/length(u)
            }
         }
      }
      return(r)
}

mnb<-0
mnbr<-0
Lb<-0
Ub<-0
```

```
LUb<-0


for (i in 1:m)
{
   xh<-rgamma(nh,ah,bh)
   xd<-rgamma(nd,ad,bd[p])

   z<-c(xd,xh)
   z<-sort(z)
   c<-z
   result<-rep(0,nd+nh)
for (i in 1:(nd+nh))
{
   if (max(xd)<min(xh))
   {
      result<-0
   }
   else
   {
      result[i]<-sum(xd>=c[i])/length(xd)+sum(xh<=c[i])/length(xh)
   }
}
result

mm<-max(result)
mm

l<-1
for (i in 1:(nh+nd))
{
   if (result[l]!=mm)
   l<-l+1
   else l<-l
}
l

chat<-c[l]

   test<-j(xd,xh,300)
   tt<-sort(test)
   bmu<-mean(tt)
   #bstd<-sqrt(var(test))
   Lb[i]<-tt[8]
   Ub[i]<-tt[293]
   mnb[i]<-(Lb[i]<=Rtt[p])*(Rtt[p]<=Ub[i])
   mnbr[i]<-(Rtt[p]>Ub[i])
   LUb[i]<-Ub[i]-Lb[i]
}
mnb<-sort(mnb)
cov.bn<-mean(mnb)
cov.bnerr<-mean(mnbr)
LUb<-sort(LUb)
wid.bn<-mean(LUb)

sink("RocBootstrapouttryBPGamma-4689-3-8080", append=T)

cat("the coverage of BP bootstrap CI for R(t):", cov.bn, "\n")
cat("the width of AC adjust bootstrap(BII) CI for R(t):", wid.bn, "\n")
cat("the lower coverage err of AC adjust bootstrap(BII) CI for R(t):", 1-
   cov.bn-cov.bnerr, "\n")
cat("the upper coverage err of AC adjust bootstrap(BII) CI for R(t):",
   cov.bnerr, "\n")
```

```
}

###code for Delta method, normal case###
alpha<-0.05
k<-qnorm(1-alpha/2)

nd<-50
nh<-50
m<-3000

uh<-0
sh<-sqrt(1)

ud<-c(0.8484,1.4071,2.1682,2.7927)
sd<-sqrt(0.5)

#ud<-c(1.2540,2.1843,3.4247,4.4340)
#sd<-sqrt(3)

#ud<-c(1.2815,2.4493,3.9625,5.1777)
#sd<-sqrt(5)

Rtt<-c(0.400017,0.599987,0.799994,0.899996)

mnb<-0
mnbr<-0
LUb<-0
vjhat<-0
jhat<-0

for (e in 1:4)
{
for (i in 1:m)
{
xh<-rnorm(nh,uh,sh)
xd<-rnorm(nd,ud[e],sd)

xhbar<-mean(xh)
xdbar<-mean(xd)

shhat<-sqrt(var(xh))
sdhat<-sqrt(var(xd))

ahat<-xdbar-xhbar
bhat<-sdhat/shhat

rad<-ahat^2+(bhat^2-1)*shhat^2*log(bhat^2)
srad<-sqrt(rad)

chat<-(xhbar*(bhat^2-1)-ahat+bhat*srad)/(bhat^2-1)

#chat<-(xhbar*(bhat^2-1)-ahat-bhat*srad)/(bhat^2-1)

zh<-(chat-xhbar)/shhat
zy<-dnorm(zh)
zd<-(xdbar-chat)/sdhat
zx<-dnorm(zd)

yy<-pnorm(zh)
xx<-pnorm(zd)

jhat[i]<-xx+yy-1
```

```
part1<-(sdhat^2/nd)*(sdhat^(-1)*zx+(shhat^(-1)*zy-sdhat^(-1)*zx)*((-
   1+ahat*bhat*srad^(-1))/(bhat^2-1)))^2
#part1<-(sdhat^2/nd)*(sdhat^(-1)*zx+(shhat^(-1)*zy-sdhat^(-1)*zx)*((-1-
   ahat*bhat*srad^(-1))/(bhat^2-1)))^2

part2<-(shhat^2/nh)*((-1)*shhat^(-1)*zy+(shhat^(-1)*zy-sdhat^(-
   1)*zx)*((bhat^2+(-1)*ahat*bhat*srad^(-1))/(bhat^2-1)))^2
#part2<-(shhat^2/nh)*((-1)*shhat^(-1)*zy+(shhat^(-1)*zy-sdhat^(-
   1)*zx)*((bhat^2-(-1)*ahat*bhat*srad^(-1))/(bhat^2-1)))^2

part3<-((sdhat^2)/(2*(nd-1)))*((-1)*zd*sdhat^(-1)*zx+(shhat^(-1)*zy-sdhat^(-
   1)*zx)*(((2*ahat*bhat)/((bhat^2-1)^2*shhat))-
((((-bhat^2-1)*srad)/((bhat^2-1)^2*shhat))+((sdhat*bhat*srad^(-1))/(bhat^2-
   1))*(log(bhat^2)+1-bhat^2))))^2
#part3<-((sdhat^2)/(2*(nd-1)))*((-1)*zd*sdhat^(-1)*zx+(shhat^(-1)*zy-sdhat^(-
   1)*zx)*(((2*ahat*bhat)/((bhat^2-1)^2*shhat))+
#((((-bhat^2-1)*srad)/((bhat^2-1)^2*shhat))+((sdhat*bhat*srad^(-1))/(bhat^2-
   1))*(log(bhat^2)+1-bhat^2))))^2

part4<-(shhat^2/(2*(nh-1)))*((-1)*zh*shhat^(-1)*zy+(shhat^(-1)*zy-sdhat^(-
   1)*zx)*((((-2)*ahat*bhat^2)/((bhat^2-1)^2*shhat))+
(((bhat*(bhat^2+1)*srad)/((bhat^2-1)^2*shhat))-((shhat*bhat*srad^(-1))/(bhat^2-
   1))*(log(bhat^2)+bhat^2-1))))^2
#part4<-(shhat^2/(2*(nh-1)))*((-1)*zh*shhat^(-1)*zy+(shhat^(-1)*zy-sdhat^(-
   1)*zx)*((((-2)*ahat*bhat^2)/((bhat^2-1)^2*shhat))-
#(((bhat*(bhat^2+1)*srad)/((bhat^2-1)^2*shhat))-((shhat*bhat*srad^(-
   1))/(bhat^2-1))*(log(bhat^2)+bhat^2-1))))^2


vjhat[i]<-sqrt(part1+part2+part3+part4)


mnb[i]<-((jhat[i]-k*vjhat[i])<=Rtt[e])*(Rtt[e]<=(jhat[i]+k*vjhat[i]))
mnbr[i]<-(Rtt[e]>(jhat[i]+k*vjhat[i]))
LUb[i]<-2*k*vjhat[i]

}
mnb<-sort(mnb)
cov.bn<-mean(mnb)
cov.bnerr<-mean(mnbr)
LUb<-sort(LUb)
wid.bn<-mean(LUb)

sink("DeltaN-0.5-5050-test", append=T)

cat("the coverage of AC adjust bootstrap CI for R(t):", cov.bn, "\n")
cat("the width of AC adjust bootstrap(BII) CI for R(t):", wid.bn, "\n")
cat("the lower coverage err of AC adjust bootstrap(BII) CI for R(t):", 1-
   cov.bn-cov.bnerr, "\n")
cat("the upper coverage err of AC adjust bootstrap(BII) CI for R(t):",
   cov.bnerr, "\n")

}

###code for Delta method, normal special case###
alpha<-0.05
k<-qnorm(1-alpha/2)

nd<-50
nh<-50
m<-3000
uh<-0
```

```
sh<-sqrt(1)

#ud<-c(0.8484,1.4071,2.1682,2.7927)
sd<-sqrt(1)

Rtt<-c(0.400017,0.599987,0.799994,0.899996)
mnb<-0
mnbr<-0
LUb<-0
vjhat<-0


for (j in 1:4)
{

for (i in 1:m)
{
xh<-rnorm(nh,uh,sh)
xd<-rnorm(nd,ud,sd)

xhbar<-mean(xh)
xdbar<-mean(xd)

shhat<-sqrt(var(xh))
sdhat<-sqrt(var(xd))

chat<-(xdbar+xhbar)/2

zh<-(chat-xhbar)/shhat
zy<-dnorm(zh)
zd<-(xdbar-chat)/sdhat
zx<-dnorm(zd)

yy<-pnorm(chat,xhbar,shhat)
xx<-pnorm(chat,xdbar,sdhat)

jhat<-yy-xx
jhat

vjhat[i]<-sqrt(sqrt((((sdhat^2/nd)*(zx+0.5*(zy-zx))^2-(shhat^2/nh)*(zy+0.5*(zy-
   zx))^2
+(1/(2*(nd-1)))*(zd*zx)^2-(1/(2*(nh-1)))*(zh*zy)^2))^2))


mnb[i]<-((jhat-k*vjhat[i])<=Rtt[j])*(Rtt[j]<=(jhat+k*vjhat[i]))
mnbr[i]<-(Rtt[j]>(jhat+k*vjhat[i]))
LUb[i]<-2*k*vjhat[i]

}
mnb<-sort(mnb)
cov.bn<-mean(mnb)
cov.bnerr<-mean(mnbr)
LUb<-sort(LUb)
wid.bn<-mean(LUb)

sink("SpecialDeltaN-4689-5050", append=T)

cat("the coverage of AC adjust bootstrap CI for R(t):", cov.bn, "\n")
cat("the width of AC adjust bootstrap(BII) CI for R(t):", wid.bn, "\n")
cat("the lower coverage err of AC adjust bootstrap(BII) CI for R(t):", 1-
   cov.bn-cov.bnerr, "\n")
cat("the upper coverage err of AC adjust bootstrap(BII) CI for R(t):",
   cov.bnerr, "\n")
```

```
sink()
}

###code for Delta method, gamma case###
alpha<-0.05
k<-qnorm(1-alpha/2)

nd<-200
nh<-200
m<-3000

ah<-1.5
bh<-1

#ad<-1.5
#bd<-c(0.4027665, 0.2295403, 0.102192, 0.0504986)
#ct<-c(2.285,2.867,3.810,4.715)
#ad<-2
#bd<-c(0.6016187, 0.3616621, 0.1769386, 0.09630156)
#ct<-c(1.991,2.619,3.572,4.478)
#ad<-2.5
#bd<-c(0.8188774, 0.5064433, 0.2619157, 0.1509695)
#ct<-c(1.788,2.448,3.417,4.315)
#ad<-3
#bd<-c(1.052237, 0.6614291, 0.3547253, 0.2124167)
#ct<-c(1.643,2.324,3.298,4.195)
#Rtt<-c(0.4, 0.6, 0.8, 0.9)

mnb<-0
mnbr<-0
LUb<-0
vjhat<-0
jhat<-0

for (q in 1:4)
{
for (i in 1:m)
{
xh<-rgamma(nh,ah,bh)
xd<-rgamma(nd,ad,bd[q])

mleh<-glm(xh~1, Gamma(link = identity))
library(MASS)
ahh <- gamma.shape(mleh)$alpha
ahh
bhh<-(ahh/coef(mleh))
bhh
rhh<-1/bhh
rhh
mled<-glm(xd~1, Gamma(link = identity))
adh <- gamma.shape(mled)$alpha
adh
bdh<-(adh/coef(mled))
bdh
rdh<-1/bdh
rdh
rd<-1/bd[q]

jhat<-pgamma(ct[q],ahh,bhh)-pgamma(ct[q],adh,bdh)
jhat

f<-function(x) c(pgamma(x,ahh,bhh)-pgamma(x,adh,bdh))
```

```
solveNonlinear(f, c(jhat), c(1.5))
chat<-solveNonlinear(f, c(jhat), c(1.5))$x
chat

deltax<-digamma(ad)
deltay<-digamma(ah)

detx<-(trigamma(ad)*ad-1)*rd^(-2)
dety<-(trigamma(ah)*ah-1)*rh^(-2)

intd <- function(x) { log(x)*exp(-x/rdh)*x^(adh-1)/(rdh^adh*gamma(adh))}
intdd<-integrate(intd, lower = chat, upper = Inf)$integral

inth <- function(x) { log(x)*exp(-x/rhh)*x^(ahh-1)/(rhh^ahh*gamma(ahh))}
inthh<-integrate(inth, lower = chat, upper = Inf)$integral

mc<-exp(-chat/rhh)*chat^(ahh-1)/(rhh^ahh*gamma(ahh))-exp(-chat/rdh)*chat^(adh-
   1)/(rdh^adh*gamma(adh))
mc

#cay
intah <- function(x) {(exp( - x/rhh) * (x^(ahh - 1) * log(x)))/(rhh^ahh *
   gamma(ahh)) - (exp( - x/rhh) * x^
   (ahh - 1) * (rhh^ahh * log(rhh) * gamma(ahh) + rhh^ahh * (exp(lgamma(ahh)) *
   digamma(ahh))))/(rhh^ahh * gamma(ahh))^2}
intahh<-integrate(intah, lower = 0, upper = chat)$integral
ccay<-(-1)*mc/intahh

#cry
intrh <- function(x) {(exp( - x/rhh) * x/rhh^2 * x^(ahh - 1))/(rhh^ahh *
   gamma(ahh)) - (exp( - x/rhh) * x^(
   ahh - 1) * (rhh^(ahh - 1) * ahh * gamma(ahh)))/(rhh^ahh * gamma(ahh))^2}
intrhh<-integrate(intrh, lower = 0, upper = chat)$integral
ccry<-(-1)*mc/intrhh

#cax
intad <- function(x) { - ((exp( - x/rdh) * (x^(adh - 1) * log(x)))/(rdh^adh *
   gamma(adh)) - (exp( - x/rdh) *
   x^(adh - 1) * (rdh^adh * log(rdh) * gamma(adh) + rdh^adh * (exp(lgamma(adh))
   *
   digamma(adh))))/(rdh^adh * gamma(adh))^2)}
intadh<-integrate(intad, lower = 0, upper = chat)$integral
ccax<-(-1)*mc/intadh

#crx
intrd <- function(x) {- ((exp( - x/rdh) * x/rdh^2 * x^(adh - 1))/(rdh^adh *
   gamma(adh)) - (exp( - x/rdh) * x^
   (adh - 1) * (rdh^(adh - 1) * adh * gamma(adh)))/(rdh^adh * gamma(adh))^2)}
intrdh<-integrate(intrd, lower = 0, upper = chat)$integral
ccrx<-(-1)*mc/intrdh

varah<-ah/(rh^2*dety*nh)
varah
varad<-ad/(rd^2*detx*nd)
varad
varbh<-trigamma(ah)/(dety*nh)
varbh
varbd<-trigamma(ad)/(detx*nd)
varbd
covabh<-(-1)/(rh*dety*nh)
covabh
covabd<-(-1)/(rd*detx*nd)
covabd
```

```
part1<-((1-pgamma(chat,adh,bdh))*(deltax-log(rdh))+ccax*((pgamma(chat,adh,bdh)-
   pgamma(chat,adh-1,bdh))/rdh+dgamma(chat,ahh,bhh))
+intdd)^2*varad
part1

part2<-((adh/rdh)*(pgamma(chat,adh,bdh)-
   pgamma(chat,adh+1,bdh))+ccrx*((pgamma(chat,adh,bdh)-pgamma(chat,adh-
   1,bdh))/rdh
+dgamma(chat,ahh,bhh)))^2*varbd
part2

part3<-((1-pgamma(chat,ahh,bhh))*(deltay-log(rhh))+ccay*((pgamma(chat,ahh,bhh)-
   pgamma(chat,ahh-1,bhh))/rhh+dgamma(chat,adh,bdh))
+inthh)^2*varah
part3

part4<-((ahh/rhh)*(pgamma(chat,ahh,bhh)-
   pgamma(chat,ahh+1,bhh))+ccry*((pgamma(chat,ahh,bhh)-pgamma(chat,ahh-
   1,bhh))/rhh
+dgamma(chat,adh,bdh)))^2*varbh
part4

part5<-((1-pgamma(chat,adh,bdh))*(deltax-log(rdh))+ccax*((pgamma(chat,adh,bdh)-
   pgamma(chat,adh-1,bdh))/rdh+dgamma(chat,ahh,bhh))
+intdd)*((adh/rdh)*(pgamma(chat,adh,bdh)-
   pgamma(chat,adh+1,bdh))+ccrx*((pgamma(chat,adh,bdh)-pgamma(chat,adh-
   1,bdh))/rdh
+dgamma(chat,ahh,bhh)))*covabd
part5

part6<-((1-pgamma(chat,ahh,bhh))*(deltay-log(rhh))+ccay*((pgamma(chat,ahh,bhh)-
   pgamma(chat,ahh-1,bhh))/rhh+dgamma(chat,adh,bdh))
+inthh)*((ahh/rhh)*(pgamma(chat,ahh,bhh)-
   pgamma(chat,ahh+1,bhh))+ccry*((pgamma(chat,ahh,bhh)-pgamma(chat,ahh-
   1,bhh))/rhh
+dgamma(chat,adh,bdh)))*covabh
part6

vjhat<-sqrt(part1+part2+part3+part4+part5+part6)
vjhat

mnb[i]<-((jhat-k*vjhat)<=Rtt[q])*(Rtt[q]<=(jhat+k*vjhat))
mnbr[i]<-(Rtt[q]>(jhat+k*vjhat))
LUb[i]<-2*k*vjhat

}
mnb<-sort(mnb)
cov.bn<-mean(mnb)
cov.bnerr<-mean(mnbr)
LUb<-sort(LUb)
wid.bn<-mean(LUb)

sink("DeltaG-4689-3-2020", append=T)

cat("the coverage of AC adjust bootstrap CI for R(t):", cov.bn, "\n")
cat("the width of AC adjust bootstrap(BII) CI for R(t):", wid.bn, "\n")
cat("the lower coverage err of AC adjust bootstrap(BII) CI for R(t):", 1-
   cov.bn-cov.bnerr, "\n")
cat("the upper coverage err of AC adjust bootstrap(BII) CI for R(t):",
   cov.bnerr, "\n")

sink()
```

```
}
```

APPENDIX C: S-Plus Code for Application

```
###real application code, NB Method###
alpha<-0.05
k<-qnorm(1-alpha/2)

x<-c(48, 49, 51, 56, 67, 67, 67, 70, 70, 72, 76, 78, 81, 82, 82, 84, 89, 99,
   126, 136)
y<-c(40, 40, 46, 47, 48, 48, 49, 49, 50, 50, 50, 50, 50, 52, 52, 55, 55, 56,
   59, 62, 62,
63, 65, 66, 71, 75, 76, 78, 83, 95, 98, 102, 187)

nd<-length(x)
nh<-length(y)

j<-function(x, y, b, k)
{
   r<-rep(NA, b)
   if (b>1)
      {
         for (i in 1:b)
         {
            t<-sample(x, length(xd), replace=T)
            u<-sample(y, length(xh), replace=T)
            if (max(t)<min(u))
            {
               r[i]<-0
            }
            else
            {
               r[i]<-
   (sum(t>=chat)+k^2/2)/(length(t)+k^2)+(sum(u<=chat)+k^2/2)/(length(u)+k^2)
            }
         }
      }
      else r[1]<-sum(xd>=chat)/length(xd)+sum(xh<=chat)/length(xh)
      return(r)
}

z<-c(x,y)
z<-sort(z)
w<-z

result<-rep(0,nd+nh)

for (i in 1:(nd+nh))
{
   if (max(x)<min(y))
   {
      result<-0
   }
   else
   {
      result[i]<-
   (sum(x>=w[i])+k^2/2)/(length(x)+k^2)+(sum(y<=w[i])+k^2/2)/(length(y)+k^2)
      #result[i]<-sum(x>=w[i])/length(x)+sum(y<=w[i])/length(y)
   }
```

```
}
result

mm<-max(result)
mm

l<-1
for (i in 1:(nh+nd))
{
   if (result[l]!=mm)
   l<-l+1
   else l<-l
}
l

chat<-w[l]
chat

test<-j(x,y,300,k)
   bmu<-mean(test)
   bstd<-sqrt(var(test))
   LUb<-2*k*bstd

jt<-(sum(x>=chat)+k^2/2)/(length(x)+k^2)+(sum(y<=chat)+k^2/2)/(length(y)+k^2)
jt
bmu
LUb
L<-jt-LUb/2
U<-jt+LUb/2
L
U

###real application code, BCa method###
xd<-c(48, 49, 51, 56, 67, 67, 67, 70, 70, 72, 76, 78, 81, 82, 82, 84, 89, 99,
126, 136)
xh<-c(40, 40, 46, 47, 48, 48, 49, 49, 50, 50, 50, 50, 50, 52, 52, 55, 55, 56,
   59, 62, 62,
63, 65, 66, 71, 75, 76, 78, 83, 95, 98, 102, 187)

nd<-length(xd)
nh<-length(xh)

jackj<-function(xd, xh, group.size=1)
{
   g<-group.size
   m<-floor(length(xd)/g)+floor(length(xh)/g)
   rr<-rep(NA, m)
   for (i in 1:floor(length(xd)/g))
   {
      t<-xd[-c(((i-1)*g+1):(i*g))]
      u<-xh
      rr[i]<-(sum(t>=chat))/(length(t))+(sum(u<=chat))/(length(u))
   }
   for (i in 1:floor(length(xh)/g))
   {
      t<-xd
      u<-xh[-c(((i-1)*g+1):(i*g))]
      rr[floor(length(xd)/g)+i]<-
   (sum(t>=chat))/(length(t))+(sum(u<=chat))/(length(u))
   }
   return(rr)
}
```

```
youd<-function(xd, xh, b)
{
   r<-rep(NA, b)
   if (b>1)
      {
         for (i in 1:b)
         {
            t<-sample(xd, length(xd), replace=T)
            u<-sample(xh, length(xh), replace=T)
            if (max(t)<min(u))
            {
               r[i]<-0
            }
            else
            {
               r[i]<-sum(t>=chat)/length(t)+sum(u<=chat)/length(u)
            }
         }
      }
      #else r[1]<-sum(xd>=chat)/length(xd)+sum(xh<=chat)/length(xh)
      return(r)
}


my.limits.bca<-function(xd, xh, obs, boot.reps, probs=c(25,50,950,975)/1000,
   details=F,
z0=NULL, acceleration=NULL, group.size=NULL)
{
   if(missing(group.size)) group.size<-max(1,
   floor(length(xd)/10)+floor(length(xh)/10))
      boot.reps<-as.matrix(boot.reps)
      accel<-acceleration
      if(is.null(accel))
      {
         accel<-0
         jackvalues<-jackj(xd,xh,group.size)
         thetahat<-mean(jackvalues)
         accel<-sum((thetahat-jackvalues)^3)/(6*(sum((thetahat-
   jackvalues)^2)^(3/2)))
      }
      zprobs<-qnorm(probs)
      if(is.null(z0))
      {
         z0<-rep(0,length(obs))
         for (i in 1:length(obs))
            z0[i]<-qnorm(mean(boot.reps[, i]<obs[i], na.rm=T))
      }
      names(z0)<-names(probs)
      emp.probs<-bca.percent<-matrix(nrow=length(obs), ncol=length(probs))
      for (i in 1:length(obs))
      {
         if(z0[i]==-Inf)
         {
            emp.probs[i, ]<-rep(0, length(emp.probs[i, ]))
            bca.percent[i, ]<-rep(min(boot.reps[, i]), length(bca.percent[i,
   ]))
         }
         else if(z0==Inf||is.na(accel))
         {
            emp.probs[i, ]<-rep(1, length(emp.probs[i, ]))
            bca.percent[i, ]<-rep(max(boot.reps[, i]), length(bca.percent[i,
   ]))
```

```
            }
            else
            {
                emp.probs[i, ]<-pnorm(z0[i]+(z0[i]+zprobs)/(1-
    accel[i]*(z0[i]+zprobs)))
                bca.percent[i, ]<-quantile(boot.reps[, i], emp.probs[i, ],
    na.rm=T)
            }
        }
        dimnames(bca.percent)<-list(names(obs), paste(100*probs, "%", sep=""))
        if(details)
        {
            dimnames(emp.probs)<-dimnames(bca.percent)
            bca.percent<-list(limits=bca.percent, emp.probs=emp.probs, z0=z0,
    acceleration=accel
            , group.size=group.size)
        }
        bca.percent
}


mn3<-0
mn3r<-0
LU3<-0


    h<-c(xd,xh)
    h<-sort(h)
    c<-h
    result<-rep(0,nd+nh)
for (i in 1:(nd+nh))
{
    if (max(xd)<min(xh))
    {
        result<-0
    }
    else
    {
        result[i]<-sum(xd>=c[i])/length(xd)+sum(xh<=c[i])/length(xh)
     }
}
result

mm<-max(result)
mm

l<-1
for (i in 1:(nh+nd))
{
    if (result[l]!=mm)
    l<-l+1
    else l<-l

}
l
chat<-c[l]
chat
youd1<-sum(xd>=chat)/length(xd)+sum(xh<=chat)/length(xh)
youd1

test<-youd(xd,xh,300)
btci<-my.limits.bca(xd,xh,youd1,test,probs=c(25,50,950,975)/1000)
Lbt<-btci[1]
```

```
Ubt<-btci[4]

Lbt
Ubt

###real application code, BP method###
xd<-c(48, 49, 51, 56, 67, 67, 67, 70, 70, 72, 76, 78, 81, 82, 82, 84, 89, 99,
   126, 136)
xh<-c(40, 40, 46, 47, 48, 48, 49, 49, 50, 50, 50, 50, 50, 52, 52, 55, 55, 56,
   59, 62, 62,
63, 65, 66, 71, 75, 76, 78, 83, 95, 98, 102, 187)

nd<-length(xd)
nh<-length(xh)


j<-function(xd, xh, b)
{
   r<-rep(NA, b)
   if (b>1)
      {
         for (i in 1:b)
         {
            t<-sample(xd, length(xd), replace=T)
            u<-sample(xh, length(xh), replace=T)
            if (max(t)<min(u))
            {
               r[i]<-0
            }
            else
            {
               r[i]<-sum(t>=chat)/length(t)+sum(u<=chat)/length(u)
            }
         }
      }
      return(r)
}



   z<-c(xd,xh)
   z<-sort(z)
   c<-z
   result<-rep(0,nd+nh)
for (i in 1:(nd+nh))
{
   if (max(xd)<min(xh))
   {
      result<-0
   }
   else
   {
      result[i]<-sum(xd>=c[i])/length(xd)+sum(xh<=c[i])/length(xh)
   }
}
result

mm<-max(result)
mm

l<-1
for (i in 1:(nh+nd))
{
```

```
   if (result[l]!=mm)
   l<-l+1
   else l<-l
}
l

chat<-c[l]
youd1<-sum(xd>=chat)/length(xd)+sum(xh<=chat)/length(xh)

   test<-j(xd,xh,300)
   tt<-sort(test)
   bmu<-mean(tt)
   Lb<-(tt[7]+tt[8])/2
   Ub<-(tt[293]+tt[292])/2
youd1
Lb
Ub
```

APPENDIX D: S-Plus Code for Parameter Calculation

```
#situation for normal
#x<-seq(0,8,by=0.0001)
#x
#######################sigmads=0.5
#a<-pnorm(((sqrt(2*x^2-log(0.5))-x)/sqrt(0.5)),0,1)
#a
#b<-pnorm((2*x-sqrt(2*x^2-log(0.5))),0,1)
#b
#j<-a+b
#j
#######################sigmads=3
#c<-pnorm(((3*x-sqrt(3*x^2+6*log(3)))/(2*sqrt(3))),0,1)
#c
#d<-pnorm(((-x+sqrt(3*x^2+6*log(3)))/2),0,1)
#d
#jj<-c+d
#jj
#######################sigmads=5
#e<-pnorm(((5*x-sqrt(5*x^2+20*log(5)))/(4*sqrt(5))),0,1)
#e
#f<-pnorm(((-x+sqrt(5*x^2+20*log(5)))/4),0,1)
#f
#jjj<-e+f
#jjj
#######################sigmads=1
#g<-pnorm(x/2,0,1)
#jjjj<-2*g
#jjjj

###################for gamma distribution ad=1.5
f<-function(x) c(pgamma(x[1],1.5,1)-pgamma(x[1],1.5,x[2]), exp(-
   x[1]+x[1]*x[2])/(x[2]^1.5))
solveNonlinear(f, c( .8, 1), c(3.5,.15))
###################for gamma distribution ad=2
f<-function(x) c(pgamma(x[1],1.5,1)-pgamma(x[1],2.5,x[2]), x[1]^(-0.5)*exp(-
   x[1]+x[1]*x[2])*gamma(2)/(x[2]^2*gamma(1.5)))
solveNonlinear(f, c( .6, 1), c(3,0.8))

###################for gamma distribution ad=2.5
f<-function(x) c(pgamma(x[1],1.5,1)-pgamma(x[1],2.5,x[2]), x[1]^(-1)*exp(-
   x[1]+x[1]*x[2])*gamma(2.5)/(x[2]^2.5*gamma(1.5)))
solveNonlinear(f, c( .9, 1), c(4,.18))

###################for gamma distribution ad=3
f<-function(x) c(pgamma(x[1],1.5,1)-pgamma(x[1],3,x[2]), x[1]^(-1.5)*exp(-
   x[1]+x[1]*x[2])*gamma(3)/(x[2]^3*gamma(1.5)))
solveNonlinear(f, c( .9, 1), c(4,.25))
```