Teacher's Corner

Statistics for Applied Researchers: Bootstrap to the Rescue

Nabendu Pal
Department of Mathematics, University of Louisiana at Lafayette

Suntaree Unhapipat
Department of Mathematics, Faculty of Science, Mahidol University

Availability of latest, fast and affordable computing resources has empowered the statisticians tremendously. This has also given the applied researchers a unique edge to extend the frontier of their knowledge-base by taking advantage of sophisticated computational statistical tools where theoretical derivations of complex sampling distributions are often not required or can be bypassed. 'Bootstrap method' is one such tool which is being used widely in solving real-life problems that involve statistical inferences. This article is designed to present bootstrap in simple terms for the applied researchers with useful examples and show how it can go a long way in settling contentious issues with reasonably convincing results.

Keywords: Sampling distribution, p-value, nonparametric bootstrap, parametric bootstrap, test statistic

1. Introduction

We start this section with a simple and interesting dataset.

Example 1.1 Prostatic Intraepithelial Neoplasia (or 'PIN') is considered to be a premalignancy, or '*carcinoma in situ*,' of the prostatic glands in men. A preliminary study [see Schardt (2011)] suggests a benefit from green tea for those at risk of prostate cancer. The study involved 60 men with PIN lesions which can turn into prostate cancer. The subjects were randomly divided into two equal groups, where each individual in one group were given 600 mg a day of a green tea extract while the individuals in the other group were given a placebo. It was a double-blind study, and the results after one year are shown in the following Table 1.1. Does the data provide evidence that taking green tea extract reduces the risk of developing prostate cancer?

Table 1.1 Results of Green Tea Extract Study on Prostate Xancer

Treatment	Cancer	No Cancer
Green tea	1	29
Placebo	9	21

The analysis of the above dataset appears to be straightforward, and such examples are found in every elementary level statistics textbook. This is a 'two population proportion problem.' To put Example 1.1 in the right perspective, we define the following basic concepts:

Population-1 = all men with PIN lesions getting 600 mg of green tea extract daily for a year,

Population-2 = all men with PIN lesions getting placebo, and

p_i = proportion of men in Population-i developing prostate cancer $i = 1, 2$.

The objective of this study is to test H_0: $p_1 \geq p_2$ against H_A: $p_1 < p_2$.

We have two samples of size $n_i = 30$, $i = 1, 2$, from the above two populations. The observed random variables are X_i, i = 1, 2, where X_i = number of men developing prostate cancer in Population-i, and X_i follows Binomial (n_i, p_i).

The widely used practice suggests using the test statistic

$$\Delta_1 = \left(\hat{p}_1 - \hat{p}_2\right) \Big/ \left\{\hat{p}\left(1-\hat{p}\right)\left(n_1^{-1} + n_2^{-1}\right)\right\}^{1/2}, \qquad (1.1)$$

where $\hat{p}_i = X_i / n_i$ and $\hat{p} = \left(X_1 + X_2\right) \Big/ \left(n_1 + n_2\right)$. From the data, it is found that $\Delta_1 = -2.7713$ with $\hat{p}_1 = 0.0333$, $\hat{p}_2 = 0.3000$ and $\hat{p} = 0.1667$. Classical theory suggests that the test statistic Δ_1, under H_0 (" = "), asymptotically follows $N(0,1)$ which gives the approximate p-value as

$$p_{approx}\text{-value}\ (\Delta_1) \approx 0.0028, \qquad (1.2)$$

which is very small prompting one to reject H_0 in favor of H_A, i.e., the green tea extract seems to have significant effect in lowering the prostate cancer risk.

But wait; green tea may really be good for lowering the risk of prostate cancer, but did we check the basic assumptions before getting the approximate p-value in (1.2)?

While the binomial distribution for each X_i seems fine (assuming that the individuals were nearly identical to begin with in terms of overall physical condition, and acted independently), how justified is the assumption that Δ_1 is approximately $N(0,1)$ when $p_1 = p_2$?

By Central Limit Theorem (*CLT*), each $\hat{p}_i$ is supposed to follow $N(p_i, p_i(1 - p_i)/n_i)$ approximately for "large n_i", $i = 1, 2$. Thus, when $p_1 = p_2 = p$

(say), using consistency of $\hat{p}$ and the *CLT* for each $\hat{p}_i$, Δ_1 is supposed to follow $N(0,1)$ for large n_i's. However, Schader and Schmid (1989) have shown that approximating the true distribution of $\hat{p}_1$ (or, that of X_i) by a suitable normal distribution may require n_i to be very large, especially when p_i is close to 0 or 1. In the current context, where p_i values are unknown, we do not know how grossly we are approximating the true distribution of Δ_1 by $N(0,1)$ under H_0 ("="). Thus, the p-value in (1.2) may be very crude. The exact distribution of Δ_1 for fixed n_i's is not known analytically (or too cumbersome to write down). [We will revisit this example later about calculating the p-value in a more justifiable manner.]

Let us look at another example.

Example 1.2 Researchers suspect that drinking tea might enhance the production of interferon gamma, a molecule that helps the immune system fight bacteria, viruses and tumors. A recent study [see Kamath et al. (2003)] involved 21 healthy individuals who did not normally drink tea or coffee. Eleven of the test subjects, selected randomly, were assigned to drink five to six cups of tea each day, while the remaining ten were asked to drink the same amount of coffee. After two weeks, blood samples were collected from the test subjects and peripheral blood mononuclear cells were cultured with the antigen alkylamine ethylalamine in an enzyme-linked immune spot assay to measure the production of interferon gamma producing cells. The results are shown in the following Table 1.2. The question of interest is whether the data provide evidence that production of interferon gamma producing cells is enhanced in tea-drinkers.

Table 1.2 Results of Tea and Coffee Effect Study

Treatment	Interferon gamma producing cell count
Tea	5, 11, 13, 18, 20, 47, 48, 52, 55, 56, 58
Coffee	0, 0, 3, 11, 15, 16, 21, 21, 38, 52

Again, this is a two population problem where Population-1 and Population-2 are defined as the collection of all only tea drinkers and collection of all only coffee drinkers, respectively. Population mean (of interferon gamma producing cell count) seems a logical parameter through which the two populations can be compared. Hence μ_i is the mean of Population-i, and one would be interested in testing $H_0: \mu_1 \leq \mu_2$ against $H_A: \mu_1 > \mu_2$.

The common practice allows one to use the test statistic

$$\Delta_2 = \left(\overline{X}_1 - \overline{X}_2\right) \Big/ \left\{\left(s_1^2/n_1\right) + \left(s_2^2/n_2\right)\right\}^{1/2} \tag{1.3}$$

where $\overline{X}_i$ and s_i are the sample mean and sample standard deviation obtained from Population-i, $i = 1, 2$. Using the approximate t_{17}-distribution for Δ_2 under H_0 (" = "), the p-value is

$$p_{approx}\text{-value}\ (\Delta_2) \approx 0.0262, \tag{1.4}$$

which seems small enough (at 5% level) to reject H_0 in favor of H_A. Hence, tea-drinking appears to have a significantly better effect over coffee-drinking. (More on the above approximation of Δ_2 by a *t*-distribution, when $\mu_1 = \mu_2$, is discussed in Subsection 3.2.)

But approximating the distribution of above Δ_2, under the assumption of "$\mu_1 = \mu_2$," by a suitable *t*-distribution, rests on two assumptions – (i) the sample sizes should be "large"; or (ii) the relative frequency histograms of the two populations ought to be (nearly) normal.

Note that, here our sample sizes $n_1 = 11$, $n_2 = 10$ are anything but large. More seriously, how do we know that the observations in two samples are normally distributed? Usually, normality is assumed when the sample relative frequency histogram looks bell-shaped, and/or formal normality tests like Shapiro-Wilk test, Anderson-Darling test, accept such an assumption. However, with samples as small as we have here, such test of normality itself is not very reliable. Figures 1.1 and 1.2 show that relative frequency histograms and Q-Q plots of the two samples.

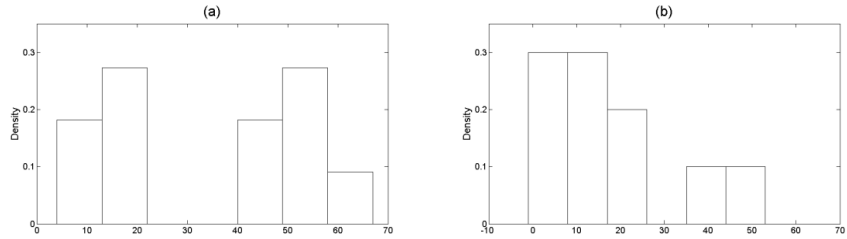

Figure 1.1 Relative Frequency Histogram: (a) Tea Drinkers, (b) Coffee Drinkers

From the above plots it seems that the normality assumption for the observations in two samples is not beyond doubts. Therefore, the sampling distribution of Δ_2 is unknown, and hence the p-value in (1.4) is questionable. So how can we draw statistical inferences when the model assumptions are not valid?

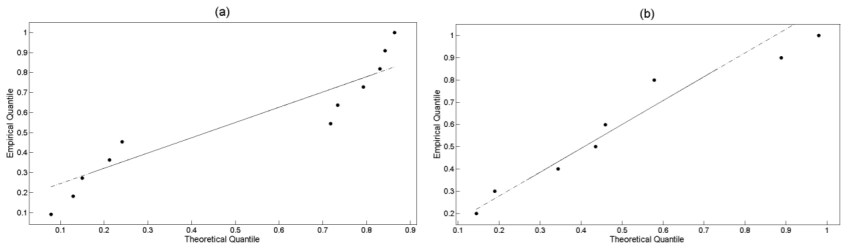

Figure 1.2 Q-Q plots: (a) tea drinkers, (b) coffee drinkers

More relevant for the above dataset (Table 1.2) would be to test $H_0 : P_1^{50} \leq P_2^{50}$ vs. $H_A : P_1^{50} > P_2^{50}$, where P_i^{50} represents the 50th percentile point (i.e., median) of Population-*i*, *i* = 1, 2, using a nonparametric approach. When a population is not symmetric, median is supposed to be a better measure of the population center than the population mean. We will revisit this comparison of medians in Section 2.

Even though the above two examples are dealing primarily with hypothesis testing, questions about model assumptions can arise in other types of inference problems as well.

Example 1.3 A blood pressure measurement consists of two numbers: the systolic pressure (*SP*), which is the maximum pressure taken when the heart is contracting, and the diastolic pressure (*DP*), which is the minimum pressure taken at the beginning of the heartbeat. Gavish, Ben-Dov and Bursztyn (2008) reported the following blood pressure dataset (Table 1.3) from a sample of 16 adults. Our goal is to find a point estimate as well as an interval estimate of ρ, the correlation coefficient between *SP* and *DP* for the population represented by the above sample.

Table 1.3 Blood Pressure Readings from Sixteen Adults

Systolic	134	115	113	123	119	118	130	116	133	112	107	110	108	105	157	154
Diastolic	87	83	77	77	69	88	76	70	91	75	71	74	69	66	103	94

It is easy to get the point estimate of ρ as $\hat{\rho} = r$ = sample correlation coefficient = 0.8568. However, the exact sampling distribution of *r* is known only if we assume bivariate normality for the above dataset. As a result, interval estimation of ρ is always a suspect. Without the normality assumption even getting the standard error (*SE*) of the above $\hat{\rho}$ is problematic.

Note that in all the above examples, it is the sampling distribution of an estimator or a test statistic (under H_0) which may look doubtful if the necessary assumptions do not hold. We will see more such examples later.

In the rest of the paper we will see how the bootstrap method, which is a heavily computational approach, can help us answer many statistical questions without worrying much about the model assumptions. Section 2 deals with nonparametric bootstrap (*NB*) approach in which no distributional assumptions are made for the dataset. However, for small samples it is often beneficial to use the parametric bootstrap (*PB*) approach which is discussed in Section 3. In Section 4 we show how bootstrap approach can be adopted for regression problems. All details have been presented with illustrative examples.

2. Nonparametric Bootstrap (*NB*)

2.1 Theory behind bootstrap

Think of a random sample of size n obtained from a population, i.e., we have independent and identically distributed (*iid*) observations $X_1, X_2,\ldots, X_n$ with cumulative distribution function (*cdf*) $F(x)$. Typically we are interested in a parameter θ which is a characteristic of the population under study. In other words, θ is a function of F, the underlying probability distribution of the random variable X whose representatives are $X_1, X_2,\ldots, X_n$. Thus, $\theta = \theta(F)$, which is unknown since F is so. For example, θ may be – (a) the mean of X, i.e., $\theta = \int x dF(x)$; (b) the population proportion of elements having values greater than c, i.e., $\theta = P(X > c) = 1 - F(c)$; (c) the population median, i.e., $F(\theta) = 0.5$, i.e., $\theta = F^{-1}(0.5)$; etc.

Our primary goal often is to estimate the parameter of interest, i.e., $\theta = \theta(F)$. Since F is unknown, it is estimated by its nonparametric maximum likelihood estimator $\hat{F}$ which is the sample distribution function, called 'Empirical Distribution Function' (*EDF*), and defined as

$$\hat{F}(x) = \left\{\sum_{i=1}^{n} I\left(X_i \le x\right)\right\} \Big/ n\,, \tag{2.1}$$

where $I(X_i \le x)$ is the indicator function taking the value 1 when $X_i \le x$, and 0 otherwise. It is seen that the indicator functions $I(X_i \le x)$, $1 \le i \le n$, are themselves Bernoulli ($F(x)$) random variables. Thus, for any fixed x, $\sum_{i=1}^{n} I\left(X_i \le x\right) \sim Binomial\left(n, F(x)\right)$. This proves easily that the nonparametric *MLE* (*NMLE*) of $F(x)$ is indeed $\hat{F}(x)$. Also,

$$Var\left(\hat{F}(x)\right) = F(x)(1 - F(x))/n\,, \tag{2.2}$$

which goes to 0 as $n \to \infty$. By Chebyshev's inequality, it is easy to show that $\hat{F}(x)$ is a consistent estimator of $F(x)$ (i.e., $\hat{F}(x)$ converges to $F(x)$ in probability, for every $x \in C(F)$ = the set of all continuity points of F, as $n \to \infty$). Further, for large n, by *CLT*

$$\hat{F}(x) \sim N\left(F(x), F(x)(1 - F(x))/n\right). \tag{2.3}$$

However, our goal is to estimate θ. By the invariance property of *MLE*, the *NMLE* of θ is

$$\hat{\theta} = \theta(\hat{F})\ . \tag{2.4}$$

The above estimation technique says that whatever characteristic θ we would like to know for the population (i.e., F) can be estimated by looking at

the corresponding characteristic of the sample (i.e., $\hat{F}$). Note that $\hat{\theta}$ is a random variable since it depends on the data. Therefore, it is important to know the precision of the estimator $\hat{\theta}$ which is measured by the standard deviation of $\hat{\theta}$, i.e., $SD(\hat{\theta}) = SD(\theta(\hat{F}))$, which is again dependent on F (since the distribution of the data is dependent on F). Hence, one needs to find an estimate of $SD(\hat{\theta})$, the so-called standard error (SE) of $\hat{\theta}$, i.e., $SE\left(\hat{\theta}\right) = \widehat{SD}\left(\hat{\theta}\right)$. More generally, one may want to know G the *cdf* of $\hat{\theta}$, i.e.,

$$G\left(y \mid F\right) = P\left(\hat{\theta} \le y \mid F\right) = P\left(\theta(\hat{F}) \le y \mid F\right) \tag{2.5}$$

The 'F' in G (and P) is explicitly written down to show that G (and P) depends on F. We will come back to $G(y|F)$ later, but first focus on obtaining $SE(\hat{\theta})$ as stated above. Without any distributional assumption, the bootstrap method provides us a mechanism to approximate $SE(\hat{\theta})$ as stated below.

Why the name "bootstrap"?

The term "bootstrapping" comes from the phrase – "to lift oneself up by his/her bootstraps (or shoe-strings)." This refers to something outrageous and impossible to accomplish. Try as hard as you can, yet you will not be able to lift yourself off the ground by pulling the shoe-strings or by tugging at pieces of leather on your boots.

However, far from alluding to its impossibility, bootstrap technique in statistical methodologies refers to the pleasantly surprising fact of its actual utility. There is an established mathematical and/or statistical theory that justifies the bootstrap method for a given dataset. Although it does not seem likely that one would be able to improve upon the estimate by reusing the same sample over and over again, bootstrapping can in fact help extract more information about the parameter estimate by using the sample repeatedly!

This is how the *NB* works: Given the original sample $\{X_1, X_2, \ldots, X_n\}$, treat it as a minipopulation, and then draw a new sample *with replacement* from this minipopulation. Call this second stage sample as $X_1^*, X_2^*, ..., X_n^*$. Note, some of the original observations (i.e., X_i's) may not show up in the bootstrap sample $\{X_1^*, X_2^*, ..., X_n^*\}$ at all, and some may show up multiple times. Anyway, based on this bootstrap sample $\{X_1^*, X_2^*, ..., X_n^*\}$ recalculate the parameter estimate as

$$\hat{\theta}^* = \theta\left(\hat{F}^*\right), \tag{2.6}$$

where $\hat{F}^*$ is the *EDF* based on $\{X_1^*, X_2^*, ..., X_n^*\}$. Repeat the above resampling from $\{X_1, X_2, \ldots, X_n\}$ a large number of times, say M times, and let the resultant

$\hat{\theta}^*$ values be $\hat{\theta}^{*(1)},...,\hat{\theta}^{*(M)}$. In other words, in the m^{th} replication, the bootstrap sample is $\left(X_1^{*(m)}, X_2^{*(m)},..., X_n^{*(m)}\right)$, based on which the parameter estimate is $\hat{\theta}^{*(m)}$, $1 \le m \le M$. The $SE(\hat{\theta})$ is taken to be

$$SE_{boot}(\hat{\theta}) = SD\left\{\hat{\theta}^{*(1)},...,\hat{\theta}^{*(M)}\right\} = \left\{\sum_{m=1}^{M}\left(\hat{\theta}^{*(m)} - \hat{\theta}^{*(\cdot)}\right)^2 \Big/ (M-1)\right\}^{1/2}, \qquad (2.7)$$

where $\hat{\theta}^{*(\cdot)} = \sum_{m=1}^{M}\hat{\theta}^{*(m)} \Big/ M$. The right side of (2.7) approximates $SD(\hat{\theta})$ fairly well even though the explicit expression of $SD(\hat{\theta})$ is either unknown, or known but complicated. By taking M sufficiently large, $SE_{boot}(\hat{\theta})$ can be made almost identical to the value $\widehat{SD}\left(\hat{\theta}\right)$. Most of the literature on bootstrapping suggest using $M = 10^3$, but here we have used $M = 10^4$.

Why does the bootstrap work?

The rationale behind bootstrap would be clear if we look at the *cdf* of $\hat{\theta}$, i.e., $G(y \mid F) = P\left(\hat{\theta} \le y \mid F\right) = P\left(\theta(\hat{F}) \le y \mid F\right)$ closely. Since F is unknown, the *MLE* of $G(y|F)$ would be $G\left(y \mid \hat{F}\right)$, which is the $G(y)$ function based on $\hat{F}$. If we draw samples from $\hat{F}$ distribution (which amounts to drawing second stage sample $X_1^*, X_2^*,..., X_n^*$ from $\{X_1, X_2,\ldots, X_n\}$) a large number of times and compute $\hat{\theta}^{*(1)},...,\hat{\theta}^{*(M)}$, then

$$G(y \mid F) = P\left(\hat{\theta} \le y \mid F\right) \approx P\left(\hat{\theta} \le y \mid \hat{F}\right) \approx (1/M)\sum_{m=1}^{M} I\left(\hat{\theta}^{*(m)} \le y\right), \qquad (2.8)$$

where $P\left(\hat{\theta} \le y \mid \hat{F}\right) = G\left(y \mid \hat{F}\right)$ and $(1/M)\sum_{m=1}^{M} I\left(\hat{\theta}^{*(m)} \le y\right) = \hat{G}^*(y)$ (say).

The above scheme tells that the *cdf* based on the bootstrap estimates $\hat{\theta}^{*(m)}$, $1 \le m \le M$, i.e., $\hat{G}^*(y)$, is an approximation to $G\left(y \mid \hat{F}\right)$, which in turn is approximating $G(y|F)$. No distributional assumption has been made so far about F. While closeness of $\hat{G}^*(y)$ to $G\left(y \mid \hat{F}\right)$ is guaranteed by taking a large M, the closeness of $G\left(y \mid \hat{F}\right)$ to $G(y|F)$ depends only on n. Therefore, success (or, failure) in approximating $G(y|F)$ by bootstrapping rests solely on n. If n is too small, then the aforementioned *NB* method provides a very crude estimate of the *cdf* of $\hat{\theta}$, in general, and $SE(\hat{\theta})$, in particular.

2.2 Mechanics of resampling

How varied are the resamples from the original sample? When we draw $\{X_1^*, X_2^*, ..., X_n^*\}$ from $\{X_1, X_2, \ldots, X_n\}$ with replacement, there are n^n possibilities, and a particular combination of a bootstrap sample appears multiple times in all n^n possibilities. Out of these n^n possibilities, the original full sample $\{X_1, X_2, \ldots, X_n\}$ appears $n!$ times. The probability p_n that a bootstrap sample is identical to the original full sample has the expression

$$p_n = n! / n^n \approx \sqrt{2\pi n}\exp(-n) \ , \tag{2.9}$$

where the right side approximation in (2.9) comes from Stirling's formula. Table 2.1 shows that even for small n, p_n is rather small.

There is another way of getting the above p_n. Using combinatorial arguments, it is seen that there are $\binom{2n-1}{n}$ distinct (and unique) possible resamples from the original sample. Out of these distinct samples, only one is identical to the original one, and hence $p_n = 1 \Big/ \binom{2n-1}{n}$, which also gives (2.9) using Stirling's approximation.

Table 2.1 Values of p_n for Various Sample Sizes

n	5	10	15	20
p_n	0.0384	3.6×10^{-4}	3.0×10^{-6}	2.3×10^{-8}

When n is too small (usually $n < 10$ is considered "too small") then bootstrap replicated samples exhibit a lot of repetitions, and hence bootstrapping may become unstable in the sense that results from one run of M resampling may vary greatly from another run of M resampling.

By drawing $\{X_1^*, X_2^*, ..., X_n^*\}$ multiple times from $\{X_1, X_2, \ldots, X_n\}$ we try to capture the sampling variation in $\hat{\theta} = \theta(\hat{F})$ through $\hat{\theta}^* = \theta(\hat{F}^*)$ values without having to derive the theoretical distribution of $\hat{\theta}$.

2.3 Applications to earlier examples

To illustrate the above bootstrap principle, let us revisit Example 1.3 first. By denoting systolic and diastolic blood pressures as U and V respectively, and writing $\boldsymbol{X} = (U, V)'$, our original sample is $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{16}\}$. (Here, resampling $\boldsymbol{X}_i^*$ from $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{16}\}$ means obtaining $\left(U_i^*, V_i^*\right)'$ as a pair, not separately resampling U_i^* from $\{U_1, \ldots, U_{16}\}$ and V_i^* from $\{V_1, \ldots, V_{16}\}$. We do not know or assume the probability distribution function (*pdf*) of $\boldsymbol{X}$ (which may or may not be

bivariate normal), and hence the distribution of $\hat{\rho}$ is completely unknown. The relative frequency histogram of the bootstrapped estimates of ρ, i.e., based on $\hat{\rho}^{*(m)}$, $1 \le m \le M$, yields the following Figure 2.1.

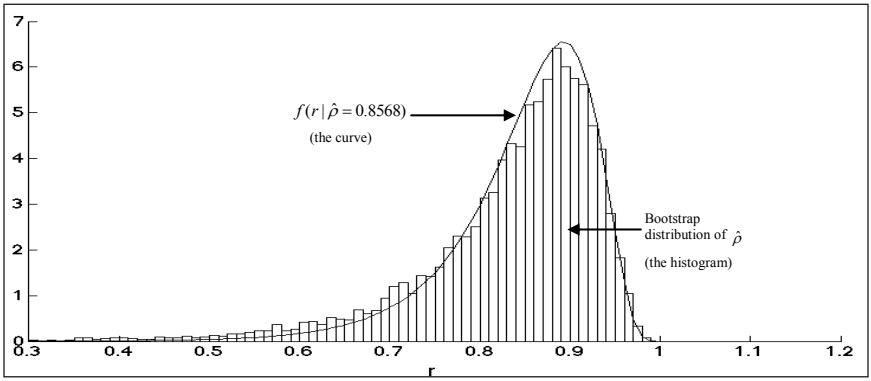

Figure 2.1 Plots of Bootstrap Distribution of $\hat{\rho}$ along with its Normal Model *pdf*

However, there are some differences which belies the visual closeness between the bootstrapped histogram of $\hat{\rho} = r$ and the estimated *pdf* of $\hat{\rho}$ based on normality. It is found that $SE_{boot}(\hat{\rho})$ = 0.1017, whereas the *SD* of $\hat{\rho}$ under the normality assumption of $\boldsymbol{X}$ is

$$SD_{normal}(\hat{\rho}) \approx \left((1-\rho^2)^2/n\right)^{1/2}$$

$$\text{i.e., } SE_{normal}(\hat{\rho}) \approx \left((1-\hat{\rho}^2)^2/n\right)^{1/2} = \sqrt{0.0044} = 0.066, \qquad (2.10)$$

which is quite different from the bootstrap estimate 0.1017.This difference in SE is due to heavier tail of the bootstrapped histogram of $\hat{\rho} = r$ than the estimated *pdf* of *r* based on normality. It is now up to the practitioner to decide which result is more plausible and applicable. Also, from $\hat{\rho}^{*(m)}$ values we get our 95% confidence interval as (0.7994, 0.9523). Bootstrap thus allows an alternative perspective and gives one a chance to '*think outside the box*.'

We now turn our attention to Examples 1.1 and 1.2 and see how bootstrap can help us answer the questions without requiring any distributional assumption. For the two hypothesis testing problems we use the significance level α = 0.05, the most commonly used value.

Example 1.1 (Revisited). Define $p_d = p_1 - p_2$. We can test H_0: $p_d \geq 0$ vs. H_A: $p_d < 0$ in two ways. Testing $p_d \geq 0$ vs. $p_d < 0$ is equivalent to testing $p_d = 0$ vs. $p_d < 0$, since the size of a test, i.e., $\text{Sup}_{H_0} P$(Type-I error), is attained at "$p_d = 0$." We give an intuitive justification behind testing only H_0: $p_d = 0$ vs. H_A: $p_d < 0$. A hypothesis testing is all about being able to differentiate between a null hypothesis value of the parameter and an alternative hypothesis value of the parameter. Therefore, distinguishing between $p_d = 0$ and $p_d < 0$ is enough for distinguishing between $p_d \geq 0$ and $p_d < 0$, since a $p_d < 0$ is closer to $p_d = 0$ than $p_d > 0$.

Approach-1: We find a left sided confidence interval (*CI*) for p_d with confidence level $(1-\alpha) = 0.95$. If the null value "$p_d = 0$" falls in that *CI*, then we retain H_0; reject H_0 otherwise. Imagine that Sample-1 (tea drinkers) is a box containing $n_1 = 30$ marbles where only $X_1 = 1$ is red, and the rest are blue. Similarly, Sample-2 (Placebo takers) is another box with $n_2 = 30$ marbles where $X_2 = 9$ are red, and the rest are blue. Draw n_i marbles with replacement from Box-i and count X_i^*, the number of red marbles in this resample. Note, $X_i^* \sim$ Binomial $(n_i, \hat{p}_i)$, $i = 1, 2,$ with $\hat{p}_i = 1/30$ and $\hat{p}_2 = 9/30$. Compute $\hat{p}_d^* = \hat{p}_1^* - \hat{p}_2^*$, where $\hat{p}_1^* = \left(X_i^* / n_i\right)$, $i = 1, 2$. Replicate this resampling process a large number of times, say M times. Thus, the $\hat{p}_d^*$ values obtained from replicated resampling are $\hat{p}_d^{*(m)}$, $1 \leq m \leq M$; and these are ordered as $\hat{p}_{d(1)}^* \leq \ldots \leq \hat{p}_{d(M)}^*$. These replicated $\hat{p}_d^{*(m)}$ values give us an idea about the sampling distribution of $\hat{p}_d$. The $(100)(1-\alpha)^{\text{th}}$ percentile value of these $\hat{p}_d^{*(m)}$ values is $\hat{p}_{du}^* = \hat{p}_{d((1-\alpha)M)}^*$, and the required $(1-\alpha)$–level left sided *CI* is $(-1, \hat{p}_{du}^*)$. Our 95% *CI* for p_d is found to be $(-1, -0.1333)$ which is also clear from the following relative frequency histogram (Figure 2.2(a)). Since H_0:$p_d = 0$ is not included in the above *CI*, H_0 is rejected at 0.05 level.

Approach-2: Instead of using the interval estimate, we can either find the critical region, or more conveniently, find the p-value based on the observed value of p_d which is $\hat{p}_d = \left(\hat{p}_1 - \hat{p}_2\right) = -0.2667$. Assume that the null hypothesis H_0:$p_d = 0$ holds. So, we combine the two samples into a single large sample of size $n = n_1 + n_2 = 60$ with $X = X_1 + X_2 = 1 + 9 = 10$. Think of a larger box with $n = 60$ marbles where $X = 10$ are red, and the remaining are blue. Now randomly divide $n = n_1 + n_2$ marbles in two smaller boxes with n_1 (= 30) and n_2 (= 30) respectively. This means we select *without replacement* 30 marbles from the larger box to put into one small box, and the remaining 30 marbles in the other small box. Let X_i^{**} = number of red marbles in Box-i with n_i marbles, $i = 1, 2$. Obtain $\hat{p}_i^{**} = \left(X_i^{**} / n_i\right)$, $i = 1, 2$, and $\hat{p}_d^{**} = \hat{p}_1^{**} - \hat{p}_2^{**}$. Repeat this process a large number of times (say, M times), and the replicated values of $\hat{p}_d^{**}$ are $\hat{p}_d^{**(m)}$, $1 \leq m \leq M$. The p-value of the bootstrap method can be found as

$$p_{boot} - value \approx \sum_{m=1}^{M} I\left(\hat{p}_d^{**(m)} \leq \hat{p}_d\right) \Big/ M \,. \tag{2.11}$$

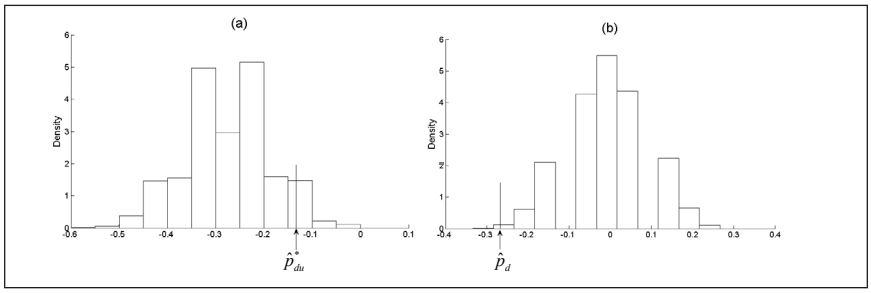

Figure 2.2 Relative Frequency Histogram of (a) $\hat{p}_d^*$ (b) $\hat{p}_d^{**}$

Our p_{boot}-value (2.11) turns out to be 0.0056, and since it is smaller than $\alpha = 0.05$, we reject H_0. Note that the results of both the above approaches agree and they must.

Remark 2.1. Notice the conceptual difference between $\hat{p}_d^*$ in Approach-1, and $\hat{p}_d^{**}$ in Approach-2. By replicating $\hat{p}_d^*$ we are approximating the distribution of $\hat{p}_d = (\hat{p}_1 - \hat{p}_2)$. Based on this approximate distribution, a one-sided *CI* for p_d is found, and if this interval admits the (minimum) null value of p_D (i.e., $p_d = 0$), then H_0 is retained. Otherwise, H_0 is rejected. On the other hand, $\hat{p}_d^{**}$ is used to approximate the distribution of $\hat{p}_d$ when H_0: $p_d = 0$ is assumed to be true. Under the approximate distribution of $\hat{p}_d^{**}$ we then find how likely is it to observe the given $\hat{p}_d$ or more extreme values (i.e. $< \hat{p}_d$). Such approximate likelihood is our p_{boot}-value = 0.0056.

Remark 2.2. Note that the p_{boot}-value for Example 1.1, which is 0.0056, is quite small. Also note p_{boot}-value (Δ_1) ≈ 0.0028 obtained from the approximate normal test. Both the p-values are implying the same decision, that is –'Reject H_0,' however, one is twice the other. So what is the difference if both methods give the same result? First of all, there is no guarantee that both the bootstrap as well as approximate method (based on normal approximation) would yield the same result always (for all possible data sets). (If they did, then they have to be theoretically identical methods.) A simulation based on $Q = 10^4$ replications and using the 0.05 significance level shows that the two methods have different sizes. Assume that H_0: $p_1 = p_2 = p$ holds (for some p). Given a value of p, in each iteration generate independent observations $X_i \sim B(n_1, p)$, $i = 1, 2$. For this dataset (X_1, X_2) carry out both the methods (the approximate normal test in (1.1) and the bootstrap method) a large number of times (say, Q times). Observe the proportion of times (out of Q iterations) each test method rejects H_0, giving the approximate size of each test. Table 2.1 provides the approximate size of each test for various p.

Table 2.1 Simulated Sizes of Two Tests for Example 1.1 (α = 0.05)

Test	*p*								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Bootstrap	0.036	0.047	0.050	0.049	0.050	0.046	0.052	0.048	0.066
Normal Approx.	0.028	0.058	0.059	0.052	0.050	0.050	0.059	0.058	0.025

It is seen from Table 2.1 that for extreme *p* the approximate normal test has lower size than the bootstrap one making it more conservative, and hence it tends to have lower power. Also, overall, the bootstrap method has size closer to α = 0.05 than the approximate normal test.

Example 1.2 (Revisited). Recall the dataset of Example 1.2 and note the fact that the individual samples, especially the first one, did not appear to be normally distributed. Consequently, the sampling distribution of the statistic Δ_2 may not follow a *t*-distribution.

For testing $H_0: \mu_1 \le \mu_2$ vs. $H_A: \mu_1 > \mu_2$ draw bootstrap samples from each sample and compute the value of Δ_2 as $\Delta_2^{*(m)} = \left(\bar{X}_1^{*(m)} - \bar{X}_2^{*(m)}\right) \Big/ \left(\sum_{i=1}^{2} s_i^{2*(m)} \Big/ n_i\right)^{1/2}$, $1 \le m \le M$. In our simulation with $M = 10^4$ iterations, the bootstrap p-value turns out to be

$$p_{boot}\text{-}value(\Delta_2) = \sum_{m=1}^{M} I\left(\Delta_2^{*(m)} > \Delta_2\right) \Big/ M = 0.0485 \qquad (2.12)$$

which is larger than that in (1.4), but still smaller than α = 0.05, thus rejects H_0.

However, since the datasets do not appear to be normal, we now test $H_0: P_1^{50} \le P_2^{50}$ vs. $H_A: P_1^{50} > P_2^{50}$ with a simple test statistic $\hat{P}_d^{50} = \left(\hat{P}_1^{50} - \hat{P}_2^{50}\right)$. For the given data, $\hat{P}_d^{50} = 31.5$. Is this $\hat{P}_d^{50}$ significantly larger than 0? Note that the exact sampling distribution of $\hat{P}_d^{50}$ under $H_0: \hat{P}_d^{50} = \hat{P}_1^{50} - \hat{P}_2^{50} = 0$ is too complicated. But we can use the bootstrap method to approximate that distribution as shown in the following histogram (Figure 2.3).The bootstrap p-value is p_{boot}-value($\hat{p}_d$) = 0.1023 which is larger enough (at 5% level) to retain H_0, i.e., in terms of median interferon cell counts we do not see significant benefit in tea drinking.

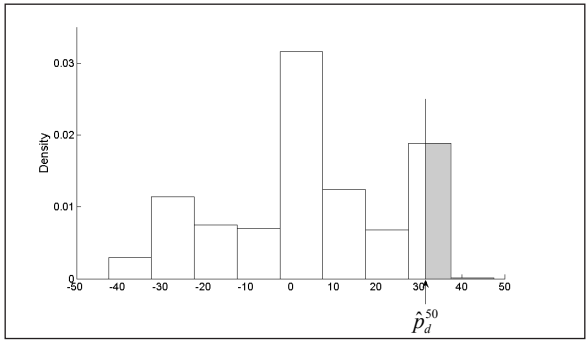

Figure 2.3 Relative Frequency Histogram of $\hat{P}_d^{50*}$ Values

Remark 2.3. When the data is nonnormal, the population median is a better measure of the center of the distribution than the population mean. The above Example 1.2 shows how the results can vary depending on the measure of the center adopted by the practitioner.

3. Parametric Bootstrap (PB)

3.1 Justification for PB

The performance of *NB* hinges on the hope that the *EDF* $\hat{F}(x)$ is a "good" estimator of the true *cdf* $F(x)$ of the random variable X under consideration. As mentioned before, $\hat{F}(x)$ is the *NMLE* of $F(x)$, and it is a consistent estimator. However, things may not be so good when n is small. With small n, $\hat{F}(x)$ is a very crude estimator of $F(x)$, and resampling from $\hat{F}(x)$ (i.e., from the original sample $\{X_1, X_2,\ldots, X_n\}$ may lead to very unstable *NB* results. Therefore, a partial compromise can be achieved through a *PB* approach.

In a *PB* approach, we assume a known shape for $F(x)$, i.e., a known parametric family of distributions where only the model parameter is unknown. To be specific, our *iid* observations $X_1, X_2,\ldots, X_n$ are following a *pdf/pmf* $f(x|\theta), \theta \in \Theta$, where the functional form 'f' is known, but the model parameter θ is unknown. Our interest lies in $\tau = \tau(\theta)$, a function of θ, which, as a special case, may be the θ itself. Our goal is to draw a suitable inference about $\tau(\theta)$ (which may be either point/interval estimation or hypothesis testing). By invariance property, the MLE of τ is

$$\hat{\tau} = \tau(\hat{\theta}) \tag{3.1}$$

where $\hat{\theta}$ is the MLE of θ based on $\{X_1, X_2,\ldots, X_n\}$ (using the known f).

If the model 'f' and/or the parametric function 'τ' are/is such (i.e., simple enough) that the sampling distribution of $\hat{\tau}$ can be obtained analytically, then we use the classical statistical theory (i.e., sufficiency, completeness, unbiasedness, Cramér–Rao inequality, Rao–Blackwell theorem, etc.) to draw inferences on τ based on $\hat{\tau}$. But what happens if $\hat{\tau}$ is intractable? Let us see the following example.

Example 3.1. Wright (1985) reported the following ground water yield (in gal/min/ft) data from two different types of wells in southwestern Virginia (USA).

Table 3.1 Ordered Yield Data from Wells with Unfractured and Fractured Rock

Rock type	Ground water yield
Unfractured	0.001, 0.003, 0.007, 0.020, 0.030, 0.040, 0.041, 0.077, 0.100, 0.454, 0.490, 1.020
Fractured	0.020, 0.031, 0.086, 0.130, 0.160, 0.160, 0.180, 0.300, 0.400, 0.440, 0.510, 0.720, 0.950

Let us focus on unfractured rock type first. Suppose our goal is to infer about $\tau = P(X > 0.5)$ = the probability of getting a minimum yield of 0.5 gal/min/ft. If we follow the nonparametric approach, then the point estimate of τ is

$$\hat{\tau}_N = \sum_{i=1}^{12} I(X_i > 0.5)/12 = 1/12 = 0.833 \tag{3.2}$$

The subscript '*N*' to $\hat{\tau}$ indicates *NMLE* of τ. Since $n = 12$ is rather "small," the estimate $\hat{\tau}_N$ is likely to be unreliable, i.e., it has a high degree of variability. But how do we get a measure of this variability? Perhaps through the SE of $\hat{\tau}_N$.

To show what happens with the *NB SE*, we carry out resampling from the above original sample *M* times to get $\hat{\tau}_N^{*(m)}$, $1 \le m \le M$. Using $M = 10^4$, the histogram is given in Figure 3.1. The *NB* approximation to the *SE* comes out to be

$$SE_{NB}\left(\hat{T}_N\right) \approx \left(\sum_{m=1}^{M} \left(\hat{\tau}_N^{*(m)} - \tau_N^{(\cdot)}\right)^2 \Big/ (M-1) \right)^{1/2} = 0.0813 \tag{3.3}$$

which appears to be quite high for an estimate of 0.0833. But there is more to it; a second resampling gives *SE* as 0.0780, which is somewhat different from (3.3). In fact, due to small *n*, repeated resampling would produce unstable *SE* values for $\hat{\tau}_N$. It can be seen that $(n\hat{\tau}_N)$ follows Binomial $(\tau, \tau\ (1-\tau))$, i.e., $\mathrm{SD}(\hat{\tau}_N) = \left(\tau(1-\tau)/n\right)^{1/2}$, i.e., $\mathrm{SE}_{est}(\hat{\tau}_N) = \left(\hat{\tau}_N(1-\hat{\tau}_N)/n\right)^{1/2} = 0.0798$, which falls right in between the two *NB SE* values mentioned above. In any case, an *SE* which is almost as large as the estimated parameter does not make the estimate too reliable.

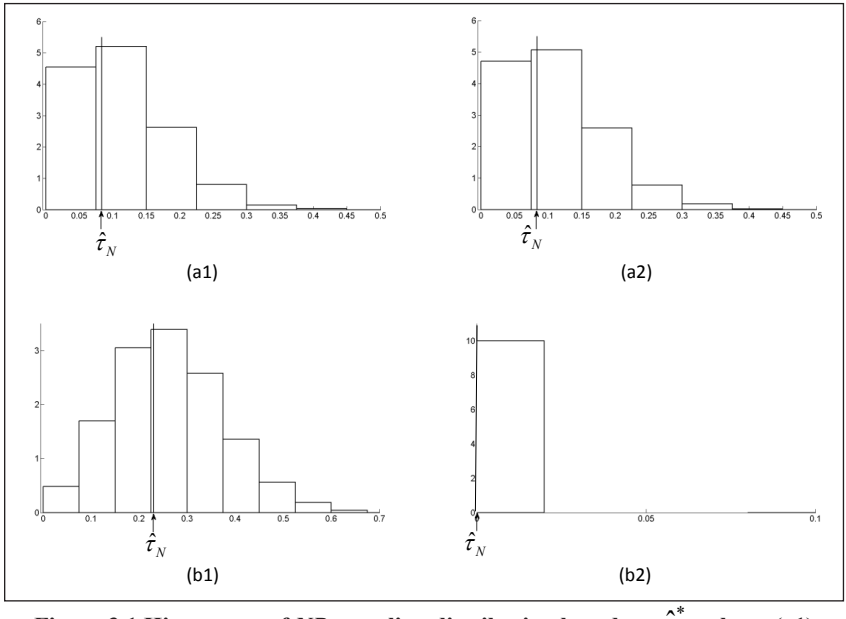

Figure 3.1 Histograms of *NB* sampling distribution based on $\hat{\tau}_N^*$ values: (a1) unfractured rock with $P(X > 0.5)$, (a2) unfractured rock with $P(X > 1.0)$, (b1) fractured rock with $P(X > 0.5)$, and (b2) fractured rock with $P(X > 1.0)$.

If we try to estimate $\tau = P(X > 1.0)$ for fractured rock, then $\hat{\tau}_N = 0.0000$ with $SE_{NB}(\hat{\tau}_N) = 0.0000$ looks absurd. This is because all observations for fractured rock type are less than 1. Does it mean that this type of wells do not give any yield more than 1? Figure 3.1 shows *NB* histograms of $\hat{\tau}_N$.

In order to address the above criticisms of the *NB* approach, we offer the *PB* approach now. First, we need to find a suitable parametric model for the given data. Typically, for hydrological problems where observations are nonnegative and tend to have a positively skewed distribution, there are some standard distributions, such as Gamma, Weibull, Lognormal, etc. Truly speaking, for the given dataset no one knows what the right model is. However, we can make an assumption that a two parameter Gamma model would be reasonably well. Note that one can argue saying that a two parameter Weibull might be appropriate too. Hence, it is just a matter of choice. We must see how well the chosen model fits the given data. Sometimes the mathematical simplicity and convenience tempt us to choose a particular model (and this is precisely the reason for using the normal model in many applications). By focusing on a smaller family of distributions we hope to gain in terms of precision in estimation (but we run the risk of mis-specifying the family).

A two-parameter Gamma model has the *pdf*

$$f(x|\delta,\sigma) = (\Gamma(\delta)\sigma^{\delta})^{-1} exp(-x/\sigma)\, x^{\delta-1},\ x > 0 \tag{3.4}$$

where $\delta > 0$ and $\sigma > 0$ are the shape and scale parameters respectively. For convenience, we call (3.4) the $G(\delta, \sigma)$ model. The MLE of (δ, σ), denoted by $(\hat{\delta}, \hat{\sigma})$, is found by the following method. First find $\hat{\delta}$ by solving the equation

$$ln\,\delta - \psi(\delta) = lnR, \tag{3.5}$$

where $\psi(x) = \partial ln\Gamma(x)/\partial x$ is the digamma function, and $R = \overline{X} / \tilde{X}$ is the ratio of sample arithmetic mean (i.e., $\overline{X} = \sum_{i=1}^{n} X_i \big/ n$) and the sample geometric mean (i.e., $\tilde{X} = \left(\prod_{i=1}^{n} X_i\right)^{1/n}$). Next, obtain $\hat{\sigma}$ by

$$\hat{\sigma} = \overline{X} / \hat{\delta}. \tag{3.6}$$

Using $(\hat{\delta}, \hat{\sigma})$, τ is now estimated by

$$\hat{\tau}_p = \int_{0.5}^{\infty} f\left(x \mid \hat{\delta}, \hat{\sigma}\right) dx = 1 - \int_{0}^{0.5} f\left(x \mid \hat{\delta}, \hat{\sigma}\right) dx. \tag{3.7}$$

The subscript '*P*' to $\hat{\tau}$ indicates 'parametric' *MLE*(*PMLE*) of τ.

Based on the fractured rock data of Example 3.1 we obtain ($\hat{\delta} = 0.434$, $\hat{\sigma} = 0.438$). The following Figure 3.2 shows the sample relative frequency histogram along with the $G(\hat{\delta}, \hat{\sigma})$ *pdf*.

To study the sampling distribution of $\hat{\tau}_p$ under the G(δ,σ) model we generate bootstrap sample $X_1^*,...,X_n^*$ (n=12) *iid* from G($\hat{\delta},\hat{\sigma}$) model, and solve the equations (3.5) - (3.7), with X_i replaced by X_1^*, $1 \le i \le n$, to obtain the *PB* estimate $\hat{\tau}_p^*$ of τ. Repeat this *PB* process M times to obtain $\hat{\tau}_p^{*(m)}$, $1 \le m \le M$. The SE($\hat{\tau}_p$) is then approximated as

$$SE_{PB}\left(\hat{\tau}_p\right) \approx \sqrt{\sum_{m=1}^{M}\left(\hat{\tau}_p^{*(m)} - \hat{\tau}_p^{*(\cdot)}\right)^2 \Big/ \left(M-1\right)} = 0.0718 \qquad (3.8)$$

where $\hat{\tau}_P^{*(\cdot)} = \sum_{m=1}^{M} \hat{\tau}_p^{*(m)} \Big/ M$. Figure 3.3 shows the *PB* distributions of $\hat{\tau}_p$.

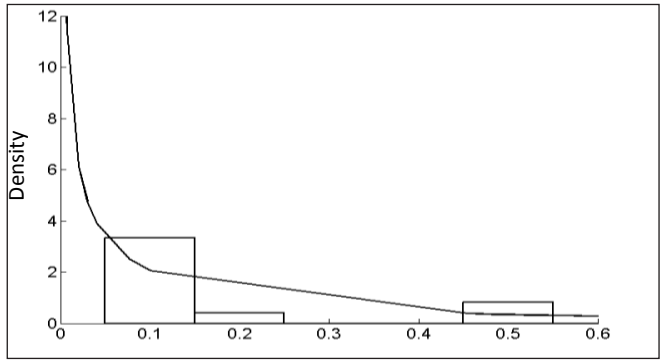

Figure 3.2 Sample Relative Frequency Histogram and the G($\hat{\delta},\hat{\sigma}$) *pdf*

Notice the difference between two *SE*s in (3.3) and (3.8). $SE_{PB}(\hat{\tau}_p)$ is (about 14%) smaller than $SE_{NB}(\hat{\tau}_N)$ because the parametric model focuses on a smaller family of Gamma distributions rather than the family of all distributions. By narrowing the focus on a smaller family of distribution the estimator $\hat{\tau}_p$ gains in precision.

Table 3.2 Summary of NB and PB Inferences on τ

Rock Type	$\tau = P(X > 0.5)$		$\tau = P(X > 1.0)$	
	NB	***PB***	***NB***	***PB***
Unfractured				
$\hat{\delta} = 0.434$ $\hat{\sigma} = 0.438$	$\hat{\tau}_{N(MLE)} = 0.083$ $SE_{NB}(\hat{\tau}_{N(MLE)})$ $= 0.080$	$\hat{\tau}_{P(MLE)} = 0.139$ $SE_{PB}(\hat{\tau}_{P(MLE)})$ $= 0.071$	$\hat{\tau}_{N(MLE)} = 0.083$ $SE_{NB}(\hat{\tau}_{N(MLE)})$ $= 0.081$	$\hat{\tau}_{P(MLE)} = 0.056$ $SE_{PB}(\hat{\tau}_{P(MLE)})$ $= 0.045$
Fractured				
$\hat{\delta} = 1.185$ $\hat{\sigma} = 0.265$	$\hat{\tau}_{N(MLE)} = 0.231$ $SE_{NB}(\hat{\tau}_{N(MLE)})$ $= 0.118$	$\hat{\tau}_{P(MLE)} = 0.199$ $SE_{PB}(\hat{\tau}_{P(MLE)})$ $= 0.090$	$\hat{\tau}_{N(MLE)} = 0.000$ $SE_{NB}(\hat{\tau}_{N(MLE)})$ $= 0.000$	$\hat{\tau}_{P(MLE)} = 0.033$ $SE_{PB}(\hat{\tau}_{P(MLE)})$ $= 0.033$

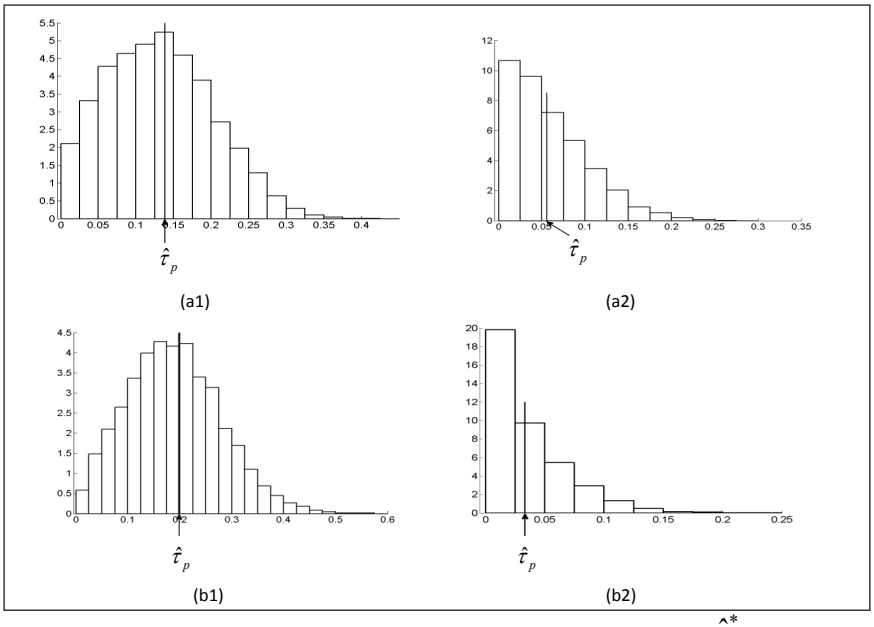

Figure 3.3 Histograms of PB Sampling Distribution based on $\hat{\tau}_p^*$ Values: (a1) Unfractured Rock with $P(X > 0.5)$, (a2) Unfractured Rock with $P(X > 1.0)$, (b1) Fractured Rock with $P(X > 0.5)$, and (b2) Fractured Rock with $P(X > 1.0)$.

3.2 PB for hypothesis testing

PB method comes very useful in hypothesis testing problems in a parametric set up where exact optimal test is either not known or has a very complicated sampling distribution. This is explained in the following.

Suppose we have *iid* observations (may be vector valued) $X_1, X_2, \ldots, X_n$ from $f(x|\theta)$, $\theta \in \Theta$, where the functional form of 'f' is assumed to be known (which puts us in a parametric set-up). Our objective is to test H_0: $\theta \in \Theta_0$ against a suitable alternative H_A: $\theta \in \Theta_A$. The classical theory suggests that the likelihood ratio test (LRT) statistic

$$\Lambda = \text{Sup}_{\theta \in \Theta_0} L(\theta \mid \boldsymbol{X}) / \text{Sup}_{\theta \in \Theta} L(\theta \mid \boldsymbol{X}) \tag{3.9}$$

be used, where the denominator is the global supremum and L is the likelihood function. Since $\Lambda \in (0,1)$, $\Delta = (-2ln\Lambda) \in (0,\infty)$. Under standard regularity conditions, the sampling distribution of Δ under H_0 can be approximated by χ^2_ν, provided n is sufficiently large, and ν = number of free parameters in Θ – number of free parameters Θ_0. Hence, for large n, one rejects H_0 if $\Delta > \chi^2_{\nu,(1-\alpha)}$ = 100 $(1-\alpha)$th percentile value of χ^2_ν distribution.

However, for n "not large," the above Chi-square distribution is not appropriate for the test statistic Δ under H_0. This gives us the opportunity to apply the *PB* method as follows.

Step 1: Obtain $\Delta = (\boldsymbol{X})$ from the given data.

Step 2: (i) Let H_0 hold, i.e., $\theta \in \Theta_0$. Find $\hat{\theta}_{RML}$ = the restricted maximum likelihood estimator of θ under H_0.

(ii) Generate artificial (bootstrap) sample $X_1^*, ..., X_n^*$ *iid* from $f\left(x \mid \hat{\theta}_{RML}\right)$. With this sample, recalculate Δ as $\Delta^* = \Delta(\boldsymbol{X}^*)$.

(iii) Repeat the above Step – 2(ii) a large number of times, say M times. The resultant Δ^* values are $\Delta^{*(m)}$, $1 \le m \le M$.

(iv) Let $\Delta^*_{(m)}$, $1 \le m \le M$, be the ordered values of $\Delta^{*(m)}$'s. The cut-off point for Δ is $\Delta_U^* = \Delta^*_{((1-\alpha)M)}$.

Step 3: Reject H_0 if $\Delta > \Delta_U^*$; retain H_0 otherwise. Alternatively, the p-value is found as

$$p_{PB}\text{-}value \approx \sum_{m=1}^{M} I\left(\Delta_m^* > \Delta\right) \Big/ M . \tag{3.10}$$

The above general *PB* method for hypothesis testing is demonstrated in a specific problem below.

Behrens-Fisher Problem

One of the oldest and most interesting problems in statistics is the Behrens-Fisher (*BF*) problem where independent samples are assumed to come from two normal populations, i.e., X_{ij}'s are independent $\sim N\left(\mu_i, \sigma_i^2\right)$, $1 \le j \le n_i$, $i = 1, 2$, where all four parameters are unknown. The objective is to test

$$H_0: \mu_1 = \mu_2 \text{ vs. } H_A: \mu_1 \neq \mu_2. \tag{3.11}$$

The minimal sufficient statistic is $\left(\bar{X}_{1\cdot}, \bar{X}_{2\cdot}, S_1, S_2\right)$, where $\bar{X}_{i\cdot} = \sum_{j=1}^{n_i} X_{ij} / n_i \sim N\left(\mu_i, \sigma_i^2 / n_i\right)$, and $\bar{X}_{i\cdot}$ is independent of $S_i = \sum_{j=1}^{n_i} \left(X_{ij} - \bar{X}_{i\cdot}\right)^2 \sim \sigma_i^2 \chi^2_{(n_i - 1)}$, $i = 1, 2$.

The literature has witnessed quite a few test methods for the above *BF* problem since 1930s [see Fisher (1935, 1941), Welch (1936, 1947, 1949), Cochran and Cox (1950), Aspin (1948), Lee and Gurland (1975), Kim and Cohen (1998), among others]. The most commonly used test is the one suggested by Welch (1949) which uses the test statistic

$$T^* = \left(\bar{X}_{1\cdot} - \bar{X}_{2\cdot}\right) \Big/ \left\{\left(s_1^2/n_1\right) + \left(s_2^2/n_2\right)\right\}^{1/2} \tag{3.12}$$

where $s_i^2 = s_i/\left(n_i - 1\right)$, i = 1, 2. The exact distribution of T^* is not easy to get; however, using Satterthwaite's (1946) approximation, the distribution of T^* under H_0 is approximated as t_k (t-distribution with k d.f), where

$$k \approx \left(\sum_{i=1}^{2}\left(s_i^2/n_i\right)\right)^2 \Big/ \left[\sum_{i=1}^{2}\left(s_i^2/n_i\right)^2/\left(n_i - 1\right)\right]. \tag{3.13}$$

Therefore, the above Welch-Satterthwaite's (*WS*) test rejects H_0 if $|T^*| > t_{k,1-(\alpha/2)}$. When k is not an integer, the critical value $t_{k,1-(\alpha/2)}$ is found by linear interpolation. Define $\lfloor k \rfloor$ as the largest integer smaller than k, then

$$t_{k,1-(\alpha/2)} \approx \left\{t_{\lfloor k \rfloor,1-(\alpha/2)}\right\} + \left\{\left(t_{\lfloor k \rfloor+1,1-(\alpha/2)} - t_{\lfloor k \rfloor,1-(\alpha/2)}\right)\left(k - \lfloor k \rfloor\right)\right\}.$$

For the above *BF* problem, $\Theta = \left\{\left(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2\right) \middle| \mu_i \in \mathbb{R}, \sigma_i^2 \in \mathbb{R}^+, i = 1, 2\right\}$. The null hypothesis restricts the parameter space to $\Theta_0 = \left\{\left(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2\right) \middle| \mu_1 = \mu_2, \mu_i \in \mathbb{R}, \sigma_i^2 \in \mathbb{R}^+, i = 1, 2\right\}$. The *PMLE* of the parameter vector $\theta = \left(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2\right)$ in Θ is $\hat{\theta}_{ML} = \left(\bar{X}_{1\cdot}, \bar{X}_{2\cdot}, S_1/n_1, S_2/n_2\right)$. When H_0 holds, then the restricted *PMLE* of θ is $\hat{\theta}_{RML} = \left(\hat{\mu}_{RML}, \hat{\mu}_{RML}, \hat{\sigma}_{1(RML)}^2, \hat{\sigma}_{2(RML)}^2\right)$ found as follows. Using $D_0 = \left(\bar{X}_{1\cdot} - \bar{X}_{2\cdot}\right)$, first find the *RML*s of σ_1^2 and σ_2^2 (i.e., $\hat{\sigma}_{1(RML)}^2$ and $\hat{\sigma}_{2(RML)}^2$) by solving (simultaneously) the system of equations

$$\begin{aligned} \sigma_1^2 &= \left(S_1/n_1\right) + \left[\left(n_2\sigma_1^2\right)^2 D_0^2 \Big/ \left(n_2\sigma_1^2 + n_1\sigma_2^2\right)^2\right], \\ \sigma_2^2 &= \left(S_2/n_2\right) + \left[\left(n_1\sigma_2^2\right)^2 D_0^2 \Big/ \left(n_2\sigma_1^2 + n_1\sigma_2^2\right)^2\right]. \end{aligned} \tag{3.14}$$

Then get $\hat{\mu}_{RML}$ by the formula

$$\hat{\mu}_{RML} = \left\{\sum_{i=1}^{2} n_i \bar{X}_{i\cdot} \Big/ \hat{\sigma}_{i(RML)}^2\right\} \Big/ \left\{\sum_{i=1}^{2} n_i \Big/ \hat{\sigma}_{i(RML)}^2\right\} . \tag{3.15}$$

Let $L\left(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2 \middle| \boldsymbol{X}\right)$ be the likelihood function, where "$\boldsymbol{X}$" represents all X_{ij}'s. Then

$$L = \prod_{i=1}^{2}\left[(2\pi)^{-n_i/2}\left(\sigma_i^2\right)^{-n_i/2} exp\left\{-\left(1/2\sigma_i^2\right)\sum_{j=1}^{n_i}\left(X_{ij}-\mu_i\right)^2\right\}\right]. \quad (3.16)$$

The *LRT* statistic is

$$\Lambda(\boldsymbol{X}) = L\left(\hat{\mu}_{RML}, \hat{\mu}_{RML}, \hat{\sigma}^2_{1(RML)}, \hat{\sigma}^2_{2(RML)}\right)\Big/ L\left(\bar{X}_{1\cdot}, \bar{X}_{2\cdot}, S_1/n_1, S_2/n_2\right). \quad (3.17)$$

Hence, our test statistic is

$$\Delta = \Delta(\boldsymbol{X}) = (-2ln\Lambda). \quad (3.18)$$

Assuming that H_0 is true, we generate X^*_{ij} from $N\left(\hat{\mu}_{RML}, \hat{\sigma}^2_{i(RML)}\right)$, $j = 1, 2, \ldots, n_i$, $i = 1, 2$. Based on this $\boldsymbol{X}^* = \left(X^*_{ij}, 1 \le j \le n_i, i = 1,2\right)$ recalculate Λ (in (3.17)) where each statistic is found by the artificial data $\mathbf{X}^*$. This gives the value of Δ as $\Delta^* = \Delta(\mathbf{X}^*) = (-2\ln\Lambda(\mathbf{X}^*))$. By replicating this Δ^* computation a large number of times (M times, say) we get the critical value for Δ (in (3.20)) as $\Delta^*_{(1-\alpha)M)}$, where $\Delta^*_{(m)}$ is the m^{th} ordered value of Δ^*.

The following Table 3.3 shows the size of the *PB* test as well as *WS* test. Here, size is free from μ (the common value of μ_1 and μ_2 under H_0), but depends only on $\gamma = \sigma_1^2/\sigma_2^2$. W.l.g. $\sigma_1^2 = 1$, and σ_2^2 is varied in our simulated size computation for a few combinations of (n_1, n_2).

Table 3.3 Simulated Size of WS and PB Tests (α = 0.05), $M = 10^4$

σ_2^2	$(n_1,n_2) = (5, 5)$		$(n_1,n_2) = (5, 10)$		$(n_1,n_2) = (10, 10)$		$(n_1,n_2) = (10, 15)$	
	WS	PB	WS	PB	WS	PB	WS	PB
10	0.044	0.051	0.042	0.048	0.043	0.048	0.047	0.051
5	0.042	0.048	0.043	0.050	0.049	0.053	0.046	0.051
2	0.041	0.046	0.047	0.049	0.040	0.045	0.048	0.051
1	0.039	0.042	0.052	0.049	0.047	0.050	0.048	0.049
0.5	0.038	0.045	0.062	0.053	0.045	0.049	0.048	0.047
0.2	0.045	0.051	0.060	0.052	0.047	0.051	0.051	0.051
0.1	0.045	0.049	0.063	0.057	0.045	0.050	0.050	0.053

Remark 3.1. Table 3.3 shows that the *PB* test attains the nominal level (α = 0.05) better than the *WS* test. The *SE* of each simulated size is approximately 0.002, which should be taken into consideration in interpreting the size values. Note that the *PB* test, for which we did not bother to find the sampling distribution of the test statistic, has size closer to α than the *WS* test whose sampling distribution has been derived in the literature with great care. To see how these two tests

fare in terms of power, Table 3.4 shows the simulated power. Here $\delta = (\mu_1 - \mu_2)$ measures deviation from H_0. Note that the power at $\delta = 0$ is nothing but size. Also *PB* is almost identical to *WS* in terms of power.

Table 3.4 Simulated Power of WS and PB Tests ($\alpha = 0.05$) with $(n_1,n_2) = (10, 10)$

σ_2^2 \ δ	0.0		0.1		0.5		1.0		1.5		2.0		3.0	
	WS	*PB*	*WS*	*PB*	*WS*	*PB*	*WS*	*PB*	*WS*	*PB*	*WS*	*PB*	*WS*	*PB*
10	0.043	0.048	0.046	0.050	0.068	0.074	0.137	0.144	0.245	0.256	0.398	0.409	0.722	0.732
5	0.045	0.049	0.046	0.051	0.090	0.096	0.212	0.225	0.407	0.424	0.640	0.656	0.939	0.945
2	0.045	0.048	0.051	0.055	0.129	0.137	0.383	0.399	0.712	0.727	0.921	0.927	0.999	0.999
1	0.042	0.044	0.051	0.055	0.177	0.185	0.549	0.562	0.884	0.891	0.986	0.987	1.0000	1.0000
0.5	0.044	0.048	0.049	0.054	0.226	0.237	0.668	0.680	0.0946	0.951	0.998	0.998	1.0000	1.0000
0.2	0.048	0.051	0.054	0.060	0.253	0.268	0.733	0.748	0.975	0.978	0.999	1.0000	1.0000	1.0000
0.1	0.049	0.052	0.055	0.059	0.281	0.292	0.772	0.781	0.983	0.985	1.000	1.000	1.0000	1.0000

4. Bootstrap for Regression Models

4.1 Basics of regression

Suppose Y is a response random variable (r.v.) of primary interest which is being explained by an explanatory variable vector $\boldsymbol{X}$ through a supposedly known function $g(\cdot)$ representing the mean response. (For example,Y is the cholesterol level of a person, and X_1 = age, X_2 = body weight, X_3 = body-mass-index (BMI), etc.) In such a case we assume the regression model

$$Y = g(\boldsymbol{X} \mid \boldsymbol{\beta}) + \varepsilon, \tag{4.1}$$

where ε is thought to be an unobservable error variable following $N(0,\sigma^2)$, distribution. The above relationship (4.1) yields $E(Y \mid \boldsymbol{X}) = g(\boldsymbol{X} \mid \boldsymbol{\beta})$ which is called regression of Y on $\boldsymbol{X}$ through the function $g(\boldsymbol{X}|\boldsymbol{\beta})$. In reality, when Y is a r.v., $E(Y \mid \boldsymbol{X})$ is unknown, but with the help of the explanatory variable $\boldsymbol{X}$ and a known functional form $g(\cdot)$ we try to explain Y, give or take some error (ε). All statistical models, whether in design of experiments, or multiple linear regression, fall under the general relationship in (4.1).

In a linear regression model we assume that $g(\boldsymbol{X}|\boldsymbol{\beta})=\boldsymbol{X'\beta}$, where the parameter vector $\boldsymbol{\beta}$ is unknown, i.e., (4.1) reduces to

$$Y=\boldsymbol{X'\beta}+\varepsilon, \tag{4.2}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_{p-1})$ and $\boldsymbol{X'} = (X_1, X_2, \ldots, X_{p-1})$.

Given a sample of n ($> p$) units from a population, we observe $(Y_i, \boldsymbol{X_i})$, $1 \le i \le n$, with the understanding

$$Y_i = \boldsymbol{X}_i'\boldsymbol{\beta} + \varepsilon_i,\ 1 \le i \le n, \tag{4.3}$$

where ε_i's are assumed to be *iid* $N(0, \sigma^2)$. (However, we will see later that the normality assumption may be skipped for a large sample.)

In general, we estimate the parameter vector $\boldsymbol{\beta}$ by minimizing the error sum of squares, which is equivalent to minimizing its square root i.e.,

$$D_2 = \left(\sum_{i=1}^{n} \varepsilon_i^2\right)^{1/2} = \left(\sum_{i=1}^{n} \left\{Y_i - g\left(\boldsymbol{X}_i \mid \boldsymbol{\beta}\right)\right\}^2\right)^{1/2}, \tag{4.4}$$

$$\text{i.e., } \hat{\boldsymbol{\beta}} = \arg\min_{\beta \in \mathbb{R}^p} D_2. \tag{4.5}$$

In the case of the linear model (4.3) the above minimization of D_2 yields

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}, \tag{4.6}$$

where $\boldsymbol{Y} = (Y_1, \ldots, Y_n)'$, $\mathbf{X} = \left(\boldsymbol{X}_1' \middle| \ldots \middle| \boldsymbol{X}_n'\right)'$. The estimator $\hat{\boldsymbol{\beta}}$ is called the least squares estimator (*LSE*), which is also the *PMLE* under the normality of the errors ε_i's. Standard statistical theory shows that under the linear model (4.3)

$$\hat{\boldsymbol{\beta}} \sim N_p\left(\boldsymbol{\beta}, \sigma^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}\right). \tag{4.7}$$

When $(\mathbf{X}'\mathbf{X})$ is not invertible, we replace its inverse by any g-inverse. Also, the realized error in (4.3) due to estimating $\boldsymbol{\beta}$ by $\hat{\boldsymbol{\beta}}$ is quantified by the 'Sum of Squares due to Error' (*SSE*) as

$$SSE = \sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{4.8}$$

where $\hat{Y}_i = \boldsymbol{X}_i'\hat{\boldsymbol{\beta}}$ which under the normality of ε_i's has the following probability distribution

$$SSE/\sigma^2 \sim \chi^2_{(n-p)}, \tag{4.9}$$

which in turn gives us an estimate of σ^2 as

$$\hat{\sigma}^2 = SSE/(n-p) = \text{Mean Square Error } (MSE). \tag{4.10}$$

Standard statistical inferences on $\boldsymbol{\beta}$ and/or σ^2 are carried out using the sampling distributions (4.7) and (4.10). But this practice has some limitations.

Why do we have to minimize the quadratic distance $\|\boldsymbol{\varepsilon}\|^2$ or the L_2–norm of $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)'$, in order to estimate $\boldsymbol{\beta}$? Or, what is the property of the estimator $\tilde{\boldsymbol{\beta}}$ which minimizes $\sum_{i=1}^{n}|e_i|$, the L_1-norm of $\boldsymbol{\varepsilon}$?

The main justification behind using the *LSE* by minimizing D_2 is the mathematical convenience, which enables us to get a closed expression of the *LSE* $\hat{\boldsymbol{\beta}}$ given in (4.6) as well as its sampling distribution given in (4.7). On the other hand, if we try to obtain $\tilde{\boldsymbol{\beta}}$ by minimizing the L_1-norm $D_1 = \sum_{i=1}^{n} |\varepsilon_i|$, then not only we do not know the expression of $\tilde{\beta}$, but also we do not have the exact sampling distribution of $\tilde{\boldsymbol{\beta}}$. However, bootstrap can be used to get an idea about the sampling distribution of $\tilde{\boldsymbol{\beta}}$ as shown in the following.

Let $D = D(Y_i, g(\boldsymbol{X}_i \mid \boldsymbol{\beta}), 1 \le i \le n)$ be any suitable metric or distance that measures the combined deviation of $g(\boldsymbol{X}_i \mid \boldsymbol{\beta})$ from Y_i's ($1 \le i \le n$). Numerically we find the optimal $\boldsymbol{\beta}$, say $\bar{\boldsymbol{\beta}}$, which minimizes D, i.e.,

$$\bar{\boldsymbol{\beta}} = \arg\min_{\beta \in \mathbb{R}^p} D. \tag{4.11}$$

(Note that when $D = D_2$, then $\bar{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}$.) The fitted values of Y due to the above estimation technique are

$$\hat{Y}_i = g(\boldsymbol{X}_i \mid \bar{\boldsymbol{\beta}}), \ 1 \le i \le n \tag{4.12}$$

The observed residuals $e_i = \left(Y_i - \hat{Y}_i\right)$, $1 \le i \le n$, give us some idea about unobservable error terms ε_i's in the model (4.1).

In order to study the sampling properties of $\bar{\boldsymbol{\beta}}$ in (4.11) through bootstrap, we can either follow the *NB* or the *PB* approach.

4.2 The NB approach to study $\bar{\boldsymbol{\beta}}$

The observed residuals $\{e_1, e_2, \ldots, e_n\}$ as defined earlier roughly represent ε_i's. So, draw a bootstrap sample of size n with replacement from $\{e_1, e_2, \ldots, e_n\}$, and let this sample be $\left\{e_1^*, e_2^*, \ldots, e_n^*\right\}$. We now create pseudo observations for the Y-variable as

$$Y_i^* = \hat{Y}_i + e_i^* = g\left(\boldsymbol{X}_i \mid \bar{\boldsymbol{\beta}}\right) + e_i^* \ , 1 \le i \le n. \tag{4.13}$$

The above Y_i^* is now paired with $\boldsymbol{X}_i$, $1 \le i \le n$, to recreate a bootstrap estimate of $\bar{\ }$ as follows. Obtain $\bar{\boldsymbol{\beta}}^*$ such that

$$\bar{\boldsymbol{\beta}}^* = \arg\min_{\beta \in \mathbb{R}^p} D\left(Y_i^*, g\left(\boldsymbol{X}_i \mid \boldsymbol{\beta}\right), 1 \le i \le n\right). \tag{4.14}$$

Repeat the above procedure (from generating e_i^*'s to getting $\bar{\boldsymbol{\beta}}^*$) a large number (say, M) times. The resultant bootstrap estimates of $\bar{\boldsymbol{\beta}}$ are $\bar{\boldsymbol{\beta}}^{*(m)}$, $1 \le m$

$\leq M$, which can be used to study sampling distribution of $\overline{\boldsymbol{\beta}}$. In particular, bias in $\overline{\boldsymbol{\beta}}$ can be estimated as

$$\widehat{Bias}\left(\overline{\boldsymbol{\beta}}\right) = \sum_{m=1}^{M}\left(\overline{\boldsymbol{\beta}}^{*(m)} - \overline{\boldsymbol{\beta}}\right)\Big/M = \overline{\boldsymbol{\beta}}^{*(\cdot)} - \overline{\boldsymbol{\beta}} \tag{4.15}$$

And the dispersion matrix of $\overline{\boldsymbol{\beta}}$ can be estimated as

$$\widehat{\text{Disp}}(\overline{\boldsymbol{\beta}}) = \sum_{m=1}^{M}\left(\overline{\boldsymbol{\beta}}^{*(m)} - \overline{\boldsymbol{\beta}}^{*(\cdot)}\right)\left(\overline{\boldsymbol{\beta}}^{*(m)} - \overline{\boldsymbol{\beta}}^{*(\cdot)}\right)'\Big/(M-1), \tag{4.16}$$

where $\overline{\boldsymbol{\beta}}^{*(\cdot)} = \sum_{m=1}^{M}\overline{\boldsymbol{\beta}}^{*(m)}\Big/M$, and this can be useful in testing a suitable hypothesis on $\overline{\boldsymbol{\beta}}$.

Note that the above *NB* approach does not need any distributional assumption about ε_i's. However, the *NB* approach works well for large n only (roughly when $n-p \geq 25$).

4.3 The PB approach to study $\overline{\boldsymbol{\beta}}$

When n is sufficiently large compared to p (usually, $n - p \geq 25$), the NB method of studying the sampling distribution of $\overline{\boldsymbol{\beta}}$ works fairly well. Note that from estimating $\boldsymbol{\beta}$ by $\overline{\boldsymbol{\beta}}$ (in (4.11)) to studying the sampling distribution properties of $\overline{\boldsymbol{\beta}}$, we do not need any distributional assumption. But when n is not sufficiently large compared to p, then the approximate sampling distribution of $\overline{\boldsymbol{\beta}}$ through *NB* approach is not very reliable. In such a case a distributional assumption may provide more reliable results (within the assumed distributional frame-work).

If we assume that ε_i's are *iid* $N(0,\sigma^2)$, then we can expect that e_i's be (roughly) normally distributed with mean 0 and variance σ^2. An estimate of σ^2 is (which is the *NMLE*)

$$\hat{\sigma}_0^2 = \sum_{i=1}^{n} e_i^2\Big/n \ . \tag{4.17}$$

(In the case of the linear model (4.2), $\hat{\sigma}_0^2$ is replaced by $\hat{\sigma}^2 = \sum_{i=1}^{n} e_i^2\Big/(n-p)$, which is slightly larger than $\hat{\sigma}_0^2$.)

For our PB method to work, we first generate e_i^{**} *iid* $N(0,\ \hat{\sigma}_0^2\ )$. Then obtain

$$Y_i^{**} = \hat{Y}_i + e_i^{**} = g\left(\boldsymbol{X}_i \mid \overline{\boldsymbol{\beta}}\right) + e_i^{**} \ , \quad 1 \leq i \leq n. \tag{4.18}$$

Then obtain $\bar{\boldsymbol{\beta}}^{**}$ by minimizing $D\left(Y_i^{**}, g\left(\boldsymbol{X}_i \mid \bar{\boldsymbol{\beta}}\right), 1 \le i \le n\right)$. By replicating this process *M* times we get $\bar{\boldsymbol{\beta}}^{**(m)}$, $1 \le m \le M$, which can now be used to study sampling properties of $\bar{\boldsymbol{\beta}}$ as done for the *NB* approach.

As a demonstration we now apply the bootstrap methods for the blood pressure dataset in Example 1.3. We follow the simple linear regression where Y = diastolic blood pressure, X = systolic blood pressure, and $\boldsymbol{\beta} = (\beta_0, \beta_1)'$.

We obtain both the *LSE* as well as the minimum L_1–norm estimate of $\boldsymbol{\beta}$, i.e., $\hat{\boldsymbol{\beta}}$ and $\tilde{\boldsymbol{\beta}}$, by minimizing $\left(\sum_{i=1}^n e_i^2\right)^{1/2}$ and $\sum_{i=1}^n |e_i|$ respectively as

$$\hat{\boldsymbol{\beta}} = (9.183, 0.575);\ \tilde{\boldsymbol{\beta}} = (7.593, 0.593). \tag{4.19}$$

The following Table 4.1 summarizes our findings. It is interesting to see that while $\hat{\boldsymbol{\beta}}$ has a smaller bias compared to $\tilde{\boldsymbol{\beta}}$, it has a larger dispersion compare to $\tilde{\boldsymbol{\beta}}$. Both the estimators are showing a negative covariance between the estimated intercept and the estimated slope. A possible reason why $\tilde{\boldsymbol{\beta}}$ has smaller dispersion than $\hat{\boldsymbol{\beta}}$ is that the L_1–norm is less effected by outliers which can affect the *LSE* in a greater way. In our blood pressure dataset there appears to be one outlier which may have affected $\hat{\boldsymbol{\beta}}$ adversely.

Table 4.1 Sampling Properties of the Estimators of $\boldsymbol{\beta}$

	Criterion			
	$\min\left(\sum_{i=1}^n e_i^2\right)^{1/2}$		$\min \sum_{i=1}^n \lvert e_i \rvert$	
	NB($\hat{\boldsymbol{\beta}}$)	*PB*($\hat{\boldsymbol{\beta}}$)	*NB*($\tilde{\boldsymbol{\beta}}$)	*PB*($\tilde{\boldsymbol{\beta}}$)
$\widehat{\text{Bias}}$	$\begin{pmatrix} 0.027 \\ 0.000 \end{pmatrix}$	$\begin{pmatrix} -0.068 \\ 0.001 \end{pmatrix}$	$\begin{pmatrix} -2.982 \\ 0.017 \end{pmatrix}$	$\begin{pmatrix} -3.079 \\ 0.019 \end{pmatrix}$
$\widehat{\text{Disp}}$	$\begin{pmatrix} 114.383 & -0.922 \\ -0.922 & 0.008 \end{pmatrix}$	$\begin{pmatrix} 128.584 & -1.038 \\ -1.038 & 0.009 \end{pmatrix}$	$\begin{pmatrix} 46.683 & -0.367 \\ -0.367 & 0.003 \end{pmatrix}$	$\begin{pmatrix} 44.683 & -0.367 \\ -0.367 & 0.003 \end{pmatrix}$

The following Figure 4.1 shows the two regression lines along with the scatter-plot.

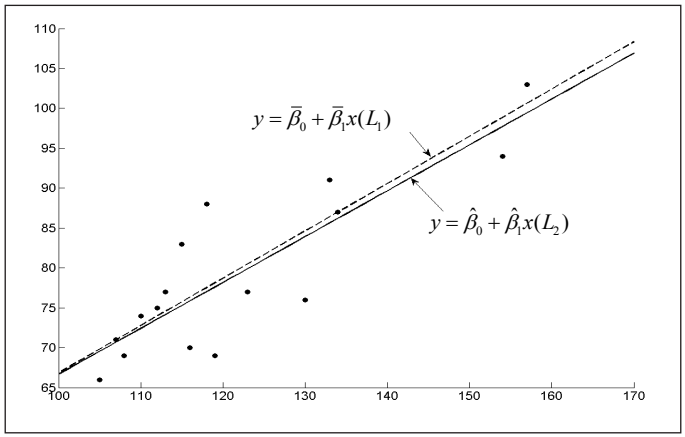

Figure 4.1 Scatter-plot along with Minimum L_1 and L_2 Norm Regression Lines

Concluding Remark. The bootstrap was first introduced by Bradley Efron in 1979 to approximate the standard error of an estimator. Its main advantage is its simplicity, although depending on applications many specialized versions of bootstrap have been developed with intricate details. For more details see Efron and Tibshirani (1993), Davison and Hinkley (1997), and the other references therein. One thing we must take note is the fact that bootstrap method is computationally intensive, and hence any small error in computations can lead to undesirable consequences. Codes of our computations are given in the Appendix.

REFERENCES

ASPIN, A. A., 1948, An examination and further development of a formula occurring in the problem of comparing two mean values, *Biometrika* 35: 28-35.

COCHRAN, W.G. AND COX, G. M., 1950, *Experimental Designs*, John Wiley, New York.

COX, D.R. AND OAKES, D., 1984, *Analysis of Survival Data*, Chapman and Hall, London (UK).

DAVISON, A.C. AND HINKLEY, D.V., 1997, Bootstrap methods and their application, *Cambridge Series in Statistical and Probabilistic Mathematics*, Cambridge University Press.

EFRON, B. AND TIBSHIRAMI, R., 1993, *An Introduction to the Bootstrap*. Chapman & Hall/CRC, Boca Raton, Florida, USA.

FISHER, R. A., 1935, The fiducial argument in statistical inference, *Annals of Eugenics*, 6: 391-398.

FISHER, R. A., 1941, The asymptotic approach to Behrens's integral with further tables for the d test of significance, *Annals of Eugenics* 11: 141-172.

GAVISH, B., BEN-DOV, I. Z. AND BURSZTYN, M., 2008, Linear relationship between systolic and diastolic blood pressure monitored over 24h: Assessment and correlates, *Journal of Hypertension* 26(2): 199-209.

KAMATH, A.B., WANG, L., DAS, H., LI, L., REINHOLD, V. N. AND BUKOWSKI, J.F., 2003, 2, Antigens in tea-beverage prime human V gamma 2V delta 2T cells in vitro and in vivo for memory and nonmemory antibacterial cytokine responses, Proceedings of the National Academy of Sciences (USA), 100, No.10, 6009-6014.

KIM, S.-H., AND COHEN, A. S., 1998, On the Behrens-Fisher problem: A review, *Journal of Educational and Behavioral Statistics* 23(4): 356-377.

LEE, A. F. S., AND GURLAND, J., 1975, Size and power of tests for equality of means of two normal population with unequal variances, *Journal of the American Statistical Association* 70: 933-941.

SCHADER, M. AND SCHMID, F., 1989, Two rules of thumb for the approximation of the binomial distribution by the normal distribution, *The American Statistician* 43: 23-24.

SCHARDT, D., 2011, What's all the fuss about green tea? Nutrition Action Healthletter, *Center for Science in the Public Interest* 10.

WELCH B. L., 1936, Specification of rules for rejecting the variable, a product with particular reference to an electric lamp problem. *Jounal of the Royal Statistical Society* 3: 29-48.

WELCH B. L., 1947, A generalization of 'Students' problem when several different population variances are involved, *Biometrika* 34: 28-35,

WELCH B. L., 1949, Further notes on Mrs. Aspin's tables, *Biometrika* 36: 243-246.

APPENDIX

A1: MATLAB code to compute the bootstrap samplingof $\hat{\rho}$ of Example 1.3 and plot the histogram as shown in Figure 2.1.

```
% original data
U = [134, 87; 115, 83; 113, 77; 123, 77;
119, 69; 118, 88; 130, 76; 116, 70; 133,
91; 112, 75; 107, 71; 110, 74; 108, 69;
105, 66; 157, 103; 154, 94];
n = length(U(:,1));
RHO = cor(U);M = 10^4;
[stats1,boot1] = bootstrp(M,@cor,U);
ord1 = sort(stats1);
p = [0.05 0.25 0.50 0.75 0.95];
meanS1 = mean(stats1);
approxSE1 = sqrt(sum((stats1 -
mean(stats1)).^2)/(M-1))
percentS1 = ord1(p*M);
RBi1 = (1/M)*sum(stats1-RHO);
r = -1:0.01:1;
f1 = (1 - (r.^2)).^((n-4)/2);
f2 = (1 - ((0.8568).^2)).^((n-1)/2);
f3 = sqrt(pi);
f4 = gamma((n-1)/2);
f5 = gamma((n-2)/2);
for i = 1:151
    k = i-1;
    f6(i,:) = ( (2*0.8568*r).^k );
    f7(i) = factorial(k);
    f8(i) = ( gamma( (n+k-1)/2 ) ).^2;
    ff(i,:) = (f6(i,:)/f7(i))*f8(i);
end
sumff = sum(ff(:,1:length(r)));
pdfr = ( (f1*f2)/(f3*f4*f5) ).*sumff;
function r = cor(U)
n = length(U(:,1));
X = U(:,1);Y = U(:,2);
r_numer = (sum(X.*Y) -
n*mean(X)*mean(Y))/n;
r_deno1 = sum((X-mean(X)).^2)/n;
r_deno2 = sum((Y-mean(Y)).^2)/n;
r_deno = sqrt(r_deno1*r_deno2);
r = r_numer/r_deno;
```

A2: MATLAB code to compute the bootstrap sampling of $\hat{p}_d^*$ and $\hat{p}_d^{**}$ and plot the histogram as shown in Figure 2.2.

```
% Approach 1
x1 = 1;x2 = 9;n1 = 30;n2 = 30;
P0 = (x1/n1) - (x2/n2);M = 10^4;
for im = 1:M
 X(1) = binornd(n1,x1/n1);
 X(2) = binornd(n2,x2/n2);
 P1(im) = (X(1)/n1) - (X(2)/n2);
end % im
P1 = sort(P1);cutP1 = P1(0.95*M);
intervalWidth = 0.075;
noIntervals = (max(P1)-min(P1))/
intervalWidth;
RR = min(P1):intervalWidth:max(P1);
ncount = histc(P1,RR);
rfreq = ncount/M;
H = rfreq/intervalWidth;
cuty = 0:0.05:max(H);
cutx = cutP1*ones(length(cuty),1);
figure();
bar(RR+(intervalWidth/2),H,'w','BarWid
th',1)
plot(cutx,cuty,'k');
BOX = [ones(10,1) zeros(10,5)];
for im = 1:M
    idx = randperm(numel(BOX));
    X2 = BOX(idx(1:30));Y2 =
    BOX(idx(31:60));
    colorX = length(find(X2 == 1));
    colorY = length(find(Y2 == 1));
    Px(im) = colorX/30;
    Py(im) = colorY/30;
    P2(im) = Px(im) - Py(im);
end
intervalWidth2 = 0.075;
noIntervals2 = (max(P2)-min(P2))/
intervalWidth2;
RR2 = min(P2):intervalWidth2:max(P2);
ncount2 = histc(P2,RR2);
rfreq2 = ncount2/M;
H2 = rfreq2/intervalWidth2;
figure();
bar(RR2+(intervalWidth2/2),H2);
cuty = 0:0.05:max(H);
cutx = P0*ones(length(cuty),1);
plot(cutx,cuty,'k');
pval = length(find(P2 <= P0))/M;
```

A3: MATLAB code to compute the bootstrap sampling of $\hat{p}_d^{50*}$ and plot the histogram as shown in Figure 2.3.

```
M = 10^4;
tea = [5 11 13 18 20 47 48 52 55 56 58];
cof = [0 0 3 11 15 16 21 21 38 52];
n1 = length(tea);n2 = length(cof);
p50_tea = median(tea);p50_cof =
median(cof);
Pd_hat = p50_tea - p50_cof;
mean_tea = mean(tea);mean_cof = mean(cof);
MeanD_hat = mean_tea - mean_cof;
Delta2=(mean_tea-mean_cof)/sqrt((var(tea)/
n1)+(var(cof)/n2));
for im = 1:M
 for in1 = 1:n1
     idx1 = randperm(numel(tea));
     X1(im,in1) = tea(idx1(1));
 end
 for in2 = 1:n2
     idx2 = randperm(numel(cof));
     X2(im,in2) = tea(idx2(1));
 end
 p50s_tea(im) = median(X1(im,:));
 p50s_cof(im) = median(X2(im,:));
 Pds_hat(im) = p50s_tea(im) - p50s_cof(im);
 means_tea(im) = mean(X1(im,:));
 means_cof(im) = mean(X2(im,:));
 MeanDs_hat(im) = means_tea(im) - means_
cof(im);
 Delta2_s(im) = (means_tea(im)-
means_cof(im))/sqrt((var(X1(im,:))/
n1)+(var(X2(im,:))/n2));
end
Pds_hat = sort(Pds_hat);
Delta2_s = sort(Delta2_s);
pval_P50 = length(find(Pds_hat > Pd_hat))/M;
pval_mean = length(find(Delta2_s >
Delta2))/M;
intWidth = 10;
noInt=(max(Pds_hat)-min(Pds_hat))/intWidth;
RR = min(Pds_hat):intWidth:max(Pds_hat);
ncount = histc(Pds_hat,RR);
rfreq = ncount/M;
H = rfreq/intWidth;
cuty = 0:0.0001:0.025;
cutx = 31.5*ones(length(cuty),1);
figure();
bar(RR+(intWidth/2),H,'w','BarWidth',1);
plot(cutx,cuty,'-k','LineWidth',1);
```

A4: MATLAB code to compute the simulated sizes of two tests for Example 1.1 in Table 2.1.

```
p = 0.1; ni = [30 30];n = sum(ni);
w = ni./n;ALPHA = 0.05;cutLRT = chi2inv(1-
ALPHA,1);
Q = 10^4;M = 10^4;
for iq = 1:Q
 xi(iq,1) = binornd(ni(1),p);
 xi(iq,2) = binornd(ni(2),p);
 p_i(iq,:) = xi(iq,:)./ni;
 phat(iq) = sum(xi(iq,:))/n;
 if (xi(iq,1) == 0) || (xi(iq,2) == 0)
 A1(iq) = 0;A2(iq,:) = 0;
 else
    A1(iq) = (phat(iq)*log(phat(iq))) +
((1-phat(iq))*log(1-phat(iq)));
    A2(iq,1) = w(1)*(
(p_i(iq,1)*log(p_i(iq,1))) + ( (1 -
p_i(iq,1))*log(1 - p_i(iq,1)) ) );
    A2(iq,2) = w(1)*(
(p_i(iq,2)*log(p_i(iq,2))) + ( (1 -
p_i(iq,2))*log(1 - p_i(iq,2)) ) );
 end
 DELTA1(iq) = n*(A1(iq) - sum(A2(iq,:)));
 DELTA(iq) = DELTA1(iq)/n;
 LRT(iq) = -2*n*DELTA(iq);
 for im = 1:M
     xi_star(im,1) =
 binornd(ni(1),phat(iq));
     xi_star(im,2) =
 binornd(ni(2),phat(iq));
     pi_star(im,:) = xi_star(im,:)./ni;
 phat_star(im) = sum(xi_star(im,:))/n;
 if (xi_star(im,1) == 0) || (xi_star(im,2)
== 0)
    A1_star(im) = 0;A2_star(im,:) = 0;
 else
 A1_star(im) = (phat_star(im)*log(phat_
star(im))) + ((1-phat_star(im))*log(1-
phat_star(im)));
    A2_star(im,1) = w(1)*( (pi_
star(im,1).*log(pi_star(im,1))) + ( (1-pi_
star(im,1)).*log(1-pi_star(im,1)) ) );
    A2_star(im,2) = w(2)*( (pi_
star(im,2).*log(pi_star(im,2))) + ( (1-pi_
star(im,2)).*log(1-pi_star(im,2)) ) );
 end
    DELTA1_star(im) = n*(A1_star(im) -
sum(A2_star(im,:)));
    DELTA_star(im) = DELTA1_star(im)/n;
 LRT_star(im) = -2*n*DELTA_star(im);
 end % end of im
 LRT_star = sort(LRT_star);
 LRTu_star = LRT_star((1-ALPHA)*M);
end % end of iq
countLRTu = length(find(LRT > LRTu_star));
PvalA = length(find(LRT > cutLRT))/Q
pvalB = countLRTu/M
```

A5: MATLAB code to compute the NB and PB simulation of $\hat{t}$ for Example 3.1 as shown in Figure 3.1 and 3.3, respectively, and the results in Table 3.2.

```
data = [0.001 0.003 0.007 0.020 0.030 0.040
0.041 0.077 0.100 0.454 0.490 1.020]; %
unfractured rock
data = [0.020 0.031 0.086 0.130 0.160 0.160
0.180 0.300 0.400 0.440 0.510 0.720 0.950];
% fractured rock
lenData = length(data);syms x;
Tn_hat = @(x) length(find(x > 1.0))/lenData;
Tn = Tn_hat(data);M = 10^4;
[Tn_star, boot] = bootstrp(M,Tn_hat,data);
SE = std(Tn_star);
T = length(find(data > 1.0))/12;
func1 = @(delta,R) log(delta) - psi(delta)
- log(mean(R)/geomean(R));
func2 = @(delta) func1(delta,data
x0 = 0.5;[delta_hat, fval] =
fsolve(func2,x0);
sigma_hat = mean(data)/delta_hat;
syms xx;
gamm = @(xx) gampdf(xx,delta_hat,sigma_
hat);
Tp = 1 - integral(gamm, 0.0001, 1.0);
for im = 1:M
 for in = 1:lenData
 Xstar(im,in) = gamrnd(delta_hat,sigma_
hat);
 end
 func3 = @(delta2)
func1(delta2,Xstar(im,:));
 [deltahat_boot, fval] =
fsolve(func3,0.01);
    sigmahat_boot = ean(Xstar(im,:))/
   deltahat_boot;
 deltaboot(im) = deltahat_boot;
 sigmaboot(im) = sigmahat_boot;
 gamm1 = @(xx) gampdf(xx,deltaboot(im),sig
maboot(im));
 TpHat(im) = 1 - integral(gamm1, 0.0001,
1.0);
end
intWidth = 0.025;
noInt=(max(TpHat)-min(TpHat))/intWidth;
RR = min(TpHat):intWidth:max(TpHat);
ncount = histc(TpHat,RR);
rfreq = ncount/M;
H = rfreq/intWidth;
figure();
bar(RR+(intWidth/2),H,'w','BarWidth',1)
cuty = 0:0.0001:max(H);
cutx = Tp*ones(length(cuty),1);
plot(cutx,cuty,'-k','LineWidth',2);
intWidth1 = 0.075;
noInt1 = (max(Tn_star)-min(Tn_star))/
intWidth1;
RR1 = min(Tn_star):intWidth1:max(Tn_star);
ncount1 = histc(Tn_star,RR1);
rfreq1 = ncount1/M;
H1 = rfreq1/intWidth1;
figure();
bar(RR1+(intWidth1/2),H1,'w','BarWidth',1);
cuty1 = 0:0.0001:max(H1);
cutx1 = Tn*ones(length(cuty1),1);
plot(cutx1,cuty1,'-k','LineWidth',2);
```

A6: FORTRAN code to compute the simulated size of WS and PB tests (Table 3.3).

```
PROGRAM BFproblem
USE random_normal_mod
USE user_set_generator
IMPLICIT NONE
REAL, parameter:: sigma1 = 1
REAL
:: sigma2, gamm
REAL, dimension(7):: sigma2_0 = (/10.0,
5.0, 2.0, 1.0, 0.5, 0.2, 0.1/)
INTEGER, parameter:: nosig = 7
INTEGER, parameter:: Q = 10**4, M = 10**4
INTEGER
     :: iq, im, isig, in1, in2
REAL*8, parameter:: alpha = 0.05
INTEGER, parameter:: n1 = 10, n2 = 15
INTEGER, dimension(2):: n = (/n1, n2/)
REAL
:: X1(Q,n1), X2(Q,n2), rn(2)
REAL, dimension(Q,2):: Xbar, S, std, Sn
REAL, dimension(Q):: WST,countWST,cutWST,D
REAL
:: kk, kk1, kk2, df_k
INTEGER, parameter:: totaln = 2
REAL
```

```
:: sig(Q, totaln), muRML(Q)
REAL, dimension(Q):: LRT, LRT1, LRT2, LRT3
REAL
:: sizeWST, sizePB
REAL::bX1(M,n1),bX2(M,n2),bXbar(M,2),bS
(M,2)
REAL::bstd(M,2),bD(M),bsig(M,2)
REAL, dimension(M)::bLRT,bLRT1,bLRT2,bLRT3
REAL, dimension(Q)
:: cutLRT, countLRT
INTEGER, PARAMETER :: MAX_SIZE = M
INTEGER, DIMENSION(1:MAX_SIZE) :: InputData
INTEGER :: ActualSize, i
call set_seeds(1)
rn(1) = real(n1)
rn(2) = real(n2)
e = etime(t)
DO isig = 1,nosig
 sigma2 = sigma2_0(isig)
 gamm = sigma1/sigma2
 DO iq = 1,Q
 DO in1 = 1,n1
X1(iq,in1) = random_
normal(0.0,sqrt(sigma1))
 END
DO
 DO in2 = 1,n2
X2(iq,in2) = random_
normal(0.0,sqrt(sigma2))
 END DO
Xbar(iq,1) = sum(X1(iq,:))/rn(1)
Xbar(iq,2) = sum(X2(iq,:))/rn(2)
S(iq,1) = sum((X1(iq,:) -
Xbar(iq,1))**2.00)
S(iq,2) = sum((X2(iq,:) -
Xbar(iq,2))**2.00)
std(iq,1) = S(iq,1)/(rn(1)-1.00)
std(iq,2) = S(iq,2)/(rn(2)-1.00)
WST(iq) = (Xbar(iq,1) - Xbar(iq,2))/&
sqrt((std(iq,1)/rn(1)) + (std(iq,2)/rn(2)))
Sn(iq,1) = (S(iq,1)**2.00)/rn(1)
Sn(iq,2) = (S(iq,2)**2.00)/rn(2)
kk1 = (Sn(iq,1) + Sn(iq,2))**2
kk2 = ((Sn(iq,1)**2.00)/(rn(1)-1.00)) +
((Sn(iq,2)**2.00)/(rn(2)-1.00))
kk = kk1/kk2
df_k = real(floor(kk))
cutWST(iq) = tcut(df_k) + ((tcut(df_k+1.00)
- tcut(df_k))*(kk - df_k))
IF (abs(WST(iq)) > cutWST(iq)) THEN
 countWST(iq) = 1
ELSE
countWST(iq) = 0
 END IF
 D(iq) = Xbar(iq,1) - Xbar(iq,2)
 sig(iq,1) = 1.0
 sig(iq,2) = 1.0
 call main(sig(iq,1), sig(iq,2), n1, n2,
S(iq,:), D(iq))
 sig(iq,1) = sig(iq,1)**2.0
 sig(iq,2) = sig(iq,2)**2.0
 muRML(iq) = (((rn(1)*Xbar(iq,1))/
sig(iq,1)) + (rn(2)*Xbar(iq,2)/sig(iq,2)))/
(rn(1)/sig(iq,1) + rn(2)/sig(iq,2))
 LRT1(iq) = sum(rn*log(rn*sig(iq,:)/
S(iq,:)))
 LRT2(iq) = sum(S(iq,:)/sig(iq,:))
 LRT3(iq) = sum((rn/
sig(iq,:))*((Xbar(iq,:)-muRML(iq))**2.0))
 LRT(iq) = LRT1(iq) + LRT2(iq) + LRT3(iq)
+ sum(rn)
 DO im = 1,M
 DO in1 = 1,n1
      bX1(im,in1) = random_normal(muRML(iq)
,sqrt(sig(iq,1)))
 END
DO
 DO in2 = 1,n2
      bX2(im,in2) = random_normal(muRML(iq)
,sqrt(sig(iq,2)))
 END DO
 bXbar(im,1) = sum(bX1(im,:))/rn(1)
 bXbar(im,2) = sum(bX2(im,:))/rn(2)
 bS(im,1) = sum((bX1(im,:) -
bXbar(im,1))**2.00)
 bS(im,2) = sum((bX2(im,:) -
bXbar(im,2))**2.00)
 bstd(im,1) = bS(im,1)/(rn(1)-1.00)
 bstd(im,2) = bS(im,2)/(rn(2)-1.00)
 bD(im) = bXbar(im,1) - bXbar(im,2)
 bsig(im,1) = 1.0
 bsig(im,2) = 1.0
 call main(bsig(im,1), bsig(im,2), n1, n2,
bS(im,:), bD(im))
 bsig(im,1) = bsig(im,1)**2.0
 bsig(im,2) = bsig(im,2)**2.0
 bmuRML(im) = (((rn(1)*bXbar(im,1))/
bsig(im,1)) + (rn(2)*bXbar(im,2)/
bsig(im,2)))/
(rn(1)/bsig(im,1) + rn(2)/bsig(im,2))
 bLRT1(im) = sum(rn*log(rn*bsig(im,:)/
bS(im,:)))
 bLRT2(im) = sum(bS(im,:)/bsig(im,:))
 bLRT3(im) = sum((rn/
bsig(im,:))*((bXbar(im,:)-
bmuRML(im))**2.0))
 bLRT(im) = bLRT1(im) + bLRT2(im) +
bLRT3(im) + sum(rn)
 END DO ! im loop
CALL Sort(bLRT, M)
cutLRT(iq) = bLRT(int(0.95*M))
IF (LRT(iq) > cutLRT(iq)) THEN
     countLRT(iq) = 1
ELSE
     countLRT(iq) = 0
 END IF
END DO ! iq loop
sizeWST = real(sum(countWST))/real(Q)
sizePB = real(sum(countLRT))/real(Q)
write(*,*) 'size of WST', sizeWST
write(*,*) 'size of PB', sizePB

END DO ! isig loop
END PROGRAM BFproblem
```

A7: FORTRAN code to compute the simulated power of WS and PB tests (Table 3.4).

```
PROGRAM BFproblem
USE random_normal_mod
USE user_set_generator
IMPLICIT NONE
REAL, parameter:: sigma1 = 1
REAL
:: sigma2, gamm, delta
REAL, dimension(7):: sigma2_0 = (/10.0,
5.0, 2.0, 1.0, 0.5, 0.2, 0.1/)
REAL, dimension(7):: delta0 = (/0.0, 0.1,
0.5, 1.0, 1.5, 2.0, 3.0/)
INTEGER, parameter:: nosig = 7, ndel = 7
INTEGER, parameter:: Q = 10**4, M = 10**4
INTEGER
:: iq, im, isig, in1, in2, idel
REAL*8, parameter:: alpha = 0.05
INTEGER, parameter:: n1 = 10, n2 = 10
REAL :: rn(2), X1(Q,n1), X2(Q,n2)
INTEGER, dimension(2) :: n = (/n1, n2/)
REAL, dimension(Q,2):: Xbar, S, std, Sn
REAL, dimension(Q):: WST,countWST,cutWST,D
REAL
:: kk, kk1, kk2, df_k
INTEGER, parameter:: totaln = 2
REAL
:: sig(Q, totaln), muRML(Q)
REAL, dimension(Q):: LRT, LRT1, LRT2, LRT3
```

```
REAL, dimension(7,7):: sizeWST, sizePB
REAL::bX1(M,n1),bX2(M,n2),bXbar(M,2),bS
(M,2)
REAL::bstd(M,2),bD(M),bsig(M,2),bmuRML(M)
REAL, dimension(M):: bLRT,bLRT1,bLRT2,bLRT3
REAL, dimension(Q):: cutLRT, countLRT
INTEGER, PARAMETER :: MAX_SIZE = M
INTEGER, DIMENSION(1:MAX_SIZE) :: InputData
INTEGER :: ActualSize, i
call set_seeds(1)
rn(1) = real(n1)
rn(2) = real(n2)
write(*,*) 'BF problem | Power Simulation'
e = etime(t)
DO isig = 1,nosig
sigma2 = sigma2_0(isig)
gamm = sigma1/sigma2
DO idel = 1, ndel
delta = delta0(idel)
DO iq = 1,Q
     DO in1 = 1,n1
     X1(iq,in1) = random_
normal(delta,sqrt(sigma1))
     END
DO
     DO in2 = 1,n2
X2(iq,in2) = random_
normal(0.0,sqrt(sigma2))
     END DO
     Xbar(iq,1) = sum(X1(iq,:))/rn(1)
     Xbar(iq,2) = sum(X2(iq,:))/rn(2)
S(iq,1) = sum((X1(iq,:) -
Xbar(iq,1))**2.00)
S(iq,2) = sum((X2(iq,:) -
Xbar(iq,2))**2.00)
     std(iq,1) = S(iq,1)/(rn(1)-1.00)
     std(iq,2) = S(iq,2)/(rn(2)-1.00)
     WST(iq) = (Xbar(iq,1) - Xbar(iq,2))/
sqrt((std(iq,1)/rn(1)) + (std(iq,2)/rn(2)))
     Sn(iq,1) = (S(iq,1)**2.00)/rn(1)
     Sn(iq,2) = (S(iq,2)**2.00)/rn(2)
     kk1 = (Sn(iq,1) + Sn(iq,2))**2
     kk2 = ((Sn(iq,1)**2.00)/(rn(1)-1.00))
+ ((Sn(iq,2)**2.00)/(rn(2)-1.00))
     kk = kk1/kk2
     df_k = real(floor(kk))
cutWST(iq) = tcut(df_k) + ((tcut(df_k+1.00)
- tcut(df_k))*(kk - df_k))
IF (abs(WST(iq)) > cutWST(iq))
THEN
countWST(iq) = 1
     ELSE
     countWST(iq) = 0
     END IF
     D(iq) = Xbar(iq,1) - Xbar(iq,2)
     sig(iq,1) = 1.0
     sig(iq,2) = 1.0
     call main(sig(iq,1), sig(iq,2), n1,
n2, S(iq,:), D(iq))
     sig(iq,1) = sig(iq,1)**2.0
     sig(iq,2) = sig(iq,2)**2.0
     muRML(iq) = (((rn(1)*Xbar(iq,1))/
sig(iq,1)) + (rn(2)*Xbar(iq,2)/sig(iq,2)))/
(rn(1)/sig(iq,1) + rn(2)/sig(iq,2))
     LRT1(iq) = sum(rn*log(rn*sig(iq,:)/
S(iq,:)))
     LRT2(iq) = sum(S(iq,:)/sig(iq,:))
     LRT3(iq) = sum((rn/
sig(iq,:))*((Xbar(iq,:)-muRML(iq))**2.0))
     LRT(iq) = LRT1(iq) + LRT2(iq) +
LRT3(iq) + sum(rn)
     DO im = 1,M
     DO in1 = 1,n1
     bX1(im,in1) = random_normal(muRML(iq),
sqrt(sig(iq,1)))
     END
DO
     DO in2 = 1,n2
     bX2(im,in2) = random_normal(muRML(iq),
sqrt(sig(iq,2)))
     END DO
     bXbar(im,1) = sum(bX1(im,:))/rn(1)
     bXbar(im,2) = sum(bX2(im,:))/rn(2)
     bS(im,1) = sum((bX1(im,:) -
bXbar(im,1))**2.00)
     bS(im,2) = sum((bX2(im,:) -
bXbar(im,2))**2.00)
     bstd(im,1) = bS(im,1)/(rn(1)-1.00)
     bstd(im,2) = bS(im,2)/(rn(2)-1.00)
     bD(im) = bXbar(im,1) - bXbar(im,2)
     bsig(im,1) = 1.0
     bsig(im,2) = 1.0
     call main(bsig(im,1), bsig(im,2), n1,
n2, bS(im,:), bD(im))
     bsig(im,1) = bsig(im,1)**2.0
     bsig(im,2) = bsig(im,2)**2.0
     bmuRML(im) = (((rn(1)*bXbar(im,1))/
bsig(im,1)) + (rn(2)*bXbar(im,2)/
bsig(im,2)))/(rn(1)/bsig(im,1) + rn(2)/
bsig(im,2))
     bLRT1(im) = sum(rn*log(rn*bsig(im,:)/
bS(im,:)))
     bLRT2(im) = sum(bS(im,:)/bsig(im,:))
     bLRT3(im) = sum((rn/
bsig(im,:))*((bXbar(im,:)-
bmuRML(im))**2.0))
     bLRT(im) = bLRT1(im) + bLRT2(im) +
bLRT3(im) + sum(rn)
     END DO ! im loop
     CALL Sort(bLRT, M)
cutLRT(iq) = bLRT(int(0.95*M))
IF (LRT(iq) > cutLRT(iq)) THEN
countLRT(iq) = 1
     ELSE
     countLRT(iq) = 0
     END IF
END DO ! iq loop
sizeWST(isig,idel) = real(sum(countWST))/
real(Q)
sizePB(isig,idel) = real(sum(countLRT))/
real(Q)
END DO ! idel loop
END DO ! isig loop
END PROGRAM BFproblem
```

A8: MATLAB code to compute the regression of the parameter β_0 and β_1 of L_1-norm and L_2-norm (Table 4.1)

```
% original data
U = [134, 87; 115, 83; 113, 77; 123, 77;
     119, 69; 118, 88; 130, 76; 116, 70;
     133, 91; 112, 75; 107, 71; 110, 74;
     108, 69; 105, 66; 157, 103; 154, 94];
n = length(U(:,1));
y = U(:,2);x = [ones(n,1) U(:,1)];
BETA1 = sum((x(:,2)-mean(x(:,2))).*(y-
mean(y)))/sum((x(:,2)-mean(x(:,2))).^2);
BETA0 = mean(y) - mean(x(:,2))*BETA1;
BETA = [BETA0; BETA1];
betahat = BETA';Yhat = x*BETA;
obsRes = y - Yhat;
syms b xx yy;
Func1 = @(b) sum(abs((y - (b(1) +
b(2)*U(:,1)))));
opts = optimset('Algorithm','interior-
point');
problem = createOptimProblem('fmincon','ob
jective',...
 Func1,'x0',[0 0],'lb',[-10 -10],'ub',[10
10],'options',opts);
gs = GlobalSearch;
[betabar,L1norm] = run(gs,problem);
M = 10^4;
sigma0 = sqrt(sum(obsRes.^2)/(n-2));
```

```
sigma0_L1 = sqrt(sum(obsRes.^2)/n);
for im = 1:M
 for in = 1:n
     ind = randperm(numel(obsRes));
     e_star(in) = obsRes(ind(end));
     e_star_PB(in) = normrnd(0,sigma0);
e_star_PB_L1(in) = normrnd(0,sigma0_L1);
 end
    Y_star(:,im) = Yhat + e_star';
    Y_star_PB(:,im) = Yhat + e_star_PB';
Y_star_PB_L1(:,im) = Yhat + e_star_PB_L1';
    meanx = mean(x(:,2));
    meany = mean(Y_star(:,im));
    meany_PB = mean(Y_star_PB(:,im));
    for i = 1:n
 BB1_NB(i) = (x(i,2) - meanx)*(Y_star(i,im)
- meany);
    BB2_NB(i) = (x(i,2) - meanx)^2;
    BB1_PB(i) = (x(i,2) - meanx)*(Y_star_
PB(i,im) - meany_PB);
 BB2_PB(i) = (x(i,2) - meanx)^2;
 end
 Bhat1_star = sum(BB1_NB)/sum(BB2_NB);
 Bhat0_star = meany - (meanx*Bhat1_star);
 Bhat_star(:,im) = [Bhat0_star; Bhat1_
star];
 Bhat1_star_PB = sum(BB1_PB)/sum(BB2_PB);
 Bhat0_star_PB = meany_PB - (meanx*Bhat1_
star_PB);
 Bhat_star_PB(:,im) = [Bhat0_star_PB;
Bhat1_star_PB];
 Func2 = @(b) sum(abs((Y_star(:,im) - (b(1)
+ b(2)*U(:,1)))));
 problem = createOptimProblem('fmincon','o
bjective',...
 Func2,'x0',[0 0],'lb',[-10 -10],'ub',[10
10],'options',opts);
 gs = GlobalSearch;
 [BB,L1norm] = run(gs,problem);
 BETAbar_star(:,im) = BB;
 Func2_PB = @(b) sum(abs((Y_star_PB_
L1(:,im) - (b(1) + b(2)*U(:,1)))));
 problem = createOptimProblem('fmincon','o
bjective',...
 Func2_PB,'x0',[0 0],'lb',[-10
-10],'ub',[10 10],'options',opts);
 gs = GlobalSearch;
 [BB_PB,L1norm_PB] = run(gs,problem);
 BETAbar_star_PB(:,im) = BB_PB;
end
for im = 1:M
 Dis(1,im) = (Bhat_star(1,im) - mean(Bhat_
star(1,:)));
 Dis(2,im) = (Bhat_star(2,im) - mean(Bhat_
star(2,:)));
 D(:,:,im) = Dis(:,im)*Dis(:,im)';
 Dis_PB(1,im) = (Bhat_star_PB(1,im) -
mean(Bhat_star_PB(1,:)));
 Dis_PB(2,im) = (Bhat_star_PB(2,im) -
mean(Bhat_star_PB(2,:)));
 D_PB(:,:,im) = Dis_PB(:,im)*Dis_PB(:,im)';
 L1Dis(1,im) = (BETAbar_star(1,im) -
mean(BETAbar_star(1,:)));
 L1Dis(2,im) = (BETAbar_star(2,im) -
mean(BETAbar_star(2,:)));
 L1D(:,:,im) = L1Dis(:,im)*L1Dis(:,im)';
 L1Dis_PB(1,im) = (BETAbar_star_PB(1,im) -
mean(BETAbar_star_PB(1,:)));
 L1Dis_PB(2,im) = (BETAbar_star_PB(2,im) -
mean(BETAbar_star_PB(2,:)));
 L1D_PB(:,:,im) = L1Dis_PB(:,im)*L1Dis_
PB(:,im)';
end
BiasBhat = [sum(Bhat_star(1,:) -
BETA(1))/M; sum(Bhat_star(2,:) -
BETA(2))/M];
BiasBhat_PB = [sum(Bhat_star_PB(1,:)
- BETA(1))/M; sum(Bhat_star_PB(2,:) -
BETA(2))/M];
BiasBbar = [sum(BETAbar_star(1,:) -
betabar(1))/M; sum(BETAbar_star(2,:) -
betabar(2))/M];
BiasBbar_PB = [sum(BETAbar_star_PB(1,:) -
betabar(1))/M; sum(BETAbar_star_PB(2,:) -
betabar(2))/M];
for i = 1:2
 for j = 1:2
DispBhat(i,j) = sum(D(i,j,:))/(M-1);
DispBhat_PB(i,j) = sum(D_PB(i,j,:))/(M-1);
DispBbar(i,j) = sum(L1D(i,j,:))/(M-1);
DispBbar_PB(i,j) = sum(L1D_PB(i,j,:))/(M-
1);
 end
end
```