Chapter 4

The Original Bootstrap Method

As shown in the previous chapter, the basic samples of data needed to calculate the confidence intervals have distributions which depart from the traditional parametric distributions. Thus, classical hypothesis-testing procedures based on strong parametric assumptions cannot be used to estimate the confidence intervals. In order to obtain results as reliable as possible, a statistical technique which is applicable regardless of the form of the data probability density function has to be utilized. In other words, this method should make no assumption about the different data distributions. One good candidate is the bootstrap method.

The bootstrap method is able to estimate measures of variability and bias. It can be employed in nonparametric or in parametric mode. The nonparametric bootstrap, which is the original bootstrap, will be described in this chapter. To fix the ideas, an example will be discussed and analyzed. Then, an algorithm that implements the method will be described. This algorithm calculates confidence intervals for nodal hourly power consumption by using the nonparametric bootstrap method.

4.1-General Principles

4.1.1-Sampling Distribution

In order to understand what bootstrapping is and how it differs from traditional parametric statistical inference, one must first be clear about the concept of the sampling distribution. Let us consider a parameter of a population probability distribution, γ, estimated by means of an estimator, Γ, using a sample drawn from that population. The sampling distribution of Γ can be thought of as the relative frequency of all possible values of Γ calculated from an infinite number of random sample of size *n* drawn from the population **[16]**. An appreciation of the factors that can influence the shape of Γ's sampling distribution is important, because it is our estimate of this sampling distribution that allows us to make inferences on γ from Γ. In our case, we will use the estimate of this sampling distribution to develop intervals in which the true value of γ lies on with a high confidence level.

Both bootstrap and traditional parametric inference seek to achieve the same goal: using limited information to estimate the sampling distribution of the chosen estimator, Γ. This estimate will be used to make inferences about a population parameter, γ. The key difference between these inferential approaches is how they obtain this sampling distribution. Whereas traditional parametric inference utilizes a priori assumptions about the shape of Γ‘s distribution, the nonparametric bootstrap is distribution free, which means that it is not dependent on a particular class of distributions. With the bootstrap method, the entire sampling distribution of Γ is estimated by relying on the fact that the sample's distribution is a good estimate of the population distribution. Traditional parametric inference depends on the assumption that the sample and the population are normally distributed.

4.1.2-Bootstrap Statistical Inference

Initiated by Efron in 1979, the basic bootstrap approach uses Monte Carlo sampling to generate an empirical estimate of the Γ‘s sampling distribution. Monte Carlo sampling builds an estimate of the sampling distribution by randomly drawing a large number of samples of size *n* from a population, and calculating for each one the associated value of the statistic Γ. The relative frequency distribution of these Γ values is an estimate of the sampling distribution for that statistic. The larger the number of samples of size *n* will be, the more accurate the relative frequency distribution of these estimates will be.

With the bootstrap method, the basic sample is treated as the population and a Monte Carlo-style procedure is conducted on it. This is done by randomly drawing a large number of 'resamples' of size *n* from this original sample (of size *n* either) with replacement. So, although each resample will have the same number of elements as the original sample, it could include some of the original data points more than once, and some not included. Therefore, each of these resamples will randomly depart from the original sample. And because the elements in these resamples vary slightly, the statistic Γ*, calculated from one of these resample will take on slightly different values. *The central assertion of the bootstrap method is that the relative frequency distribution of these* G**'s is an estimate of the sampling distribution of* G.

4.2-Procedure

The steps of the generic nonparametric bootstrap procedure can be stated more formally (Efron and Tibshirani **[15]**) as follows. Consider the case where a random sample of size *n* is drawn from an unspecified probability distribution, ω. The basic steps in the bootstrap procedure are

Step 1. Construct an empirical probability distribution, Ω, from the sample by placing a probability of 1/*n* at each point, $x_1, x_2, ..., x_n$ of the sample. This is the empirical distribution function of the sample, which is the nonparametric maximum likelihood estimate of the population distribution, ω. Now, each sample's element has the same probability to be drawn.

Step 2. From the empirical distribution function, Ω, draw a random sample of size *n* with replacement. This is a 'resample'.

Step 3. Calculate the statistic of interest, Γ, for this resample, yielding Γ*.

Step 4. Repeat steps 2 and 3 *B* times, where *B* is a large number, in order to create *B* resamples. The practical size of *B* depends on the tests to be run on the data. Typically, *B* is at least equal to 1000 when an estimate of confidence interval around Γ is required.

Step 5. Construct the relative frequency histogram from the *B* number of Γ*'s by placing a probability of 1/*B* at each point, $\Gamma^*_1, \Gamma^*_2, ..., \Gamma^*_B$. The distribution obtained is the bootstrapped estimate of the sampling distribution of Γ. This distribution can now be used to make inferences about the parameter γ.

4.2.1-Example

An example may serve to clarify this procedure. Although the sample mean may be evaluated in that case by using a parametric approach, this example provides a good decription of how the nonparametric bootstrap works and demonstrates its accuracy.

Let us consider a sample containing two hundred values generated randomly from a standard normal population N(0,1). This is the original sample. In this example, the sampling distribution of the arithmetic mean is approximately normal with a mean roughly equal to 0 and a standard deviation approximately equal to $1/\sqrt{200}$. Now, let us apply the nonparametric bootstrap method to infer the result. One thousand and five hundred resamples are drawn from the original sample, and the arithmetic mean is calculated for each resample. These calculations are performed by using S-PLUS functions as follows

Step 1. Randomly draw two hundred points from a standard normal population

gauss<-rnorm (200,0,1)

Step 2. Perform the nonparametric bootstrap study (1500 resamples)

for (i in 1:1500) boot[i]<-mean (sample (gauss,replace=T))

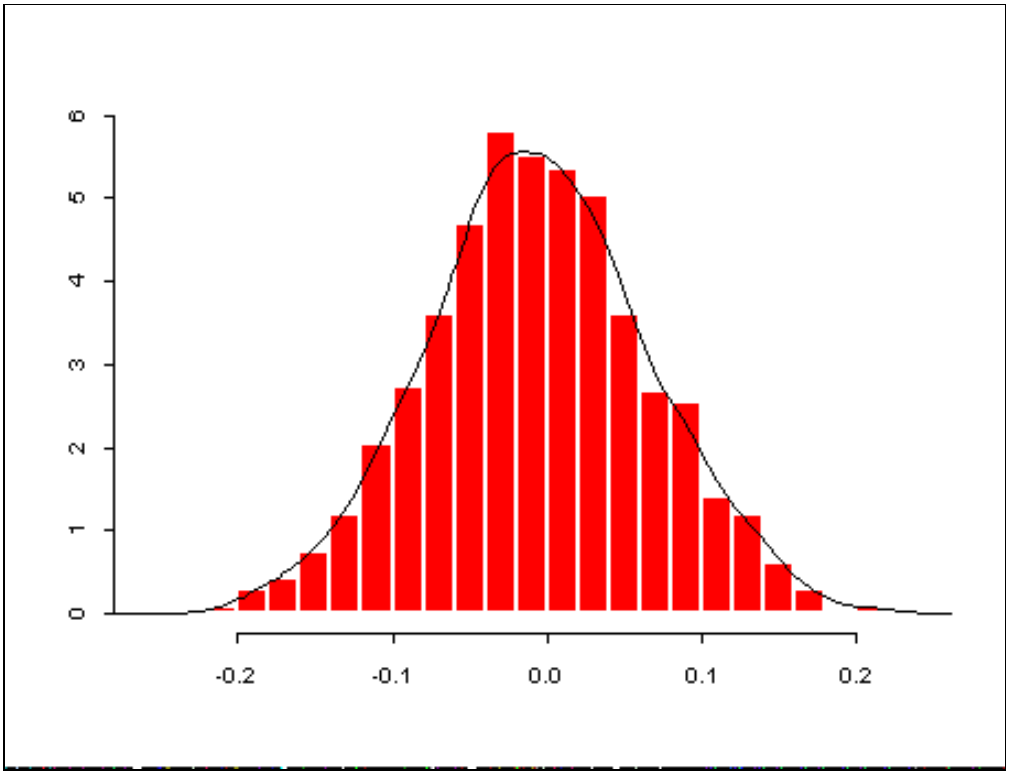


Fig 4.1 Estimate of the Mean Sampling Distribution by using the Nonparametric Bootstrap Method

It is shown from Table 4.1 that the bootstrap method provides results very close to the parametric method.

Table 4.1 Characteristics of Parametric and Bootstrapped Sampling Distributions for the Mean

Method used	Mean	Standard Deviation	Quantile 2.50%	Quantile 97.50%
Param.	0	0.0707	-0.13	0.13
Bootstrap	-0.006	0.0713	-0.144	0.136

4.2.2-Theoretical Basis

At this point, it is possible to discuss why this method works as well as it does in the above example. At root is the idea that *if the sample is a good approximation of the population, the bootstrap method will provide a good approximation of the sampling distribution of* G. The conceptual justification of the bootstrap procedure rests on the fact that (a) the sample empirical distribution function (*EDF*) is an estimate of the population distribution function (*PDF*), and (b) the random resampling mechanism with the stochastic component of the model. The theoretical justification for these analogies is based on two levels of asymptotic

Level 1. As the original sample size *n* approaches the population size, the *EDF* approaches the true distribution, *PDF*. This makes intuitive sense, in that as a sample increases in size, it contains more and more information about the population until being equal to the population.

Level 2. If the original sample's size *n* is large enough, then, as the number *B* of resamples increases to infinity, the bootstrapped estimate of the sampling distribution approaches the sampling distribution of the original statistic Γ.

Although mathematical proof of consistency is an important justification, the criterion of practicality also needs to be considered. How large an *n* and *B* are 'large enough' to provide satisfactory results? This is an empirical question that depends on the statistics to be estimated and the accuracy desired. The size of *B* is merely a computational concern because, with a looping algorithm, it is strictly a function of the computing time. However, we may consider that the improvement of the bootstrapped estimate of the sampling distribution is notable when $B > 1000$, in most cases **[15, 16]**.

4.3-Random Generator

From a basic data sample of size N = N1+N2+N3, a nonparametric bootstrap study can be carried out by building resamples and by calculating for each one the statistic considered. Every resample consists of N values randomly drawn with replacement from the basic sample. To be reliable, these calculations require to have both a basic sample representative of the population and an efficient generator of random numbers. As underlined in Chapter 2, a reliable random generator is also needed to obtain desired basic samples. Thus, it is very important to implement this generator carefully.

Usually, in each computer, a library routine which is called a 'random number generator' can be found. It turns out, however, that this system-supplied generator usually does not meet the

need of the bootstrap method. Hence, a random generator is going to be proposed. Two classes of random generators can be distinguished depending on the following factors: reliability and speed. The more reliable the generator is, the more slowly it runs. Taking into consideration these two factors and our specific needs, the congruential random generator described below was chosen **[19, 20]**.

At the beginning of the routine, the user sets an integer number called the seed number. This number, X_0, is used by the program to initialize the generator. This number being chosen, a new random number is then generated by using the general formula

$$X_i \leftarrow \left(a\, X_{i-1} + c\right) \bmod M \qquad \textbf{(1)}$$

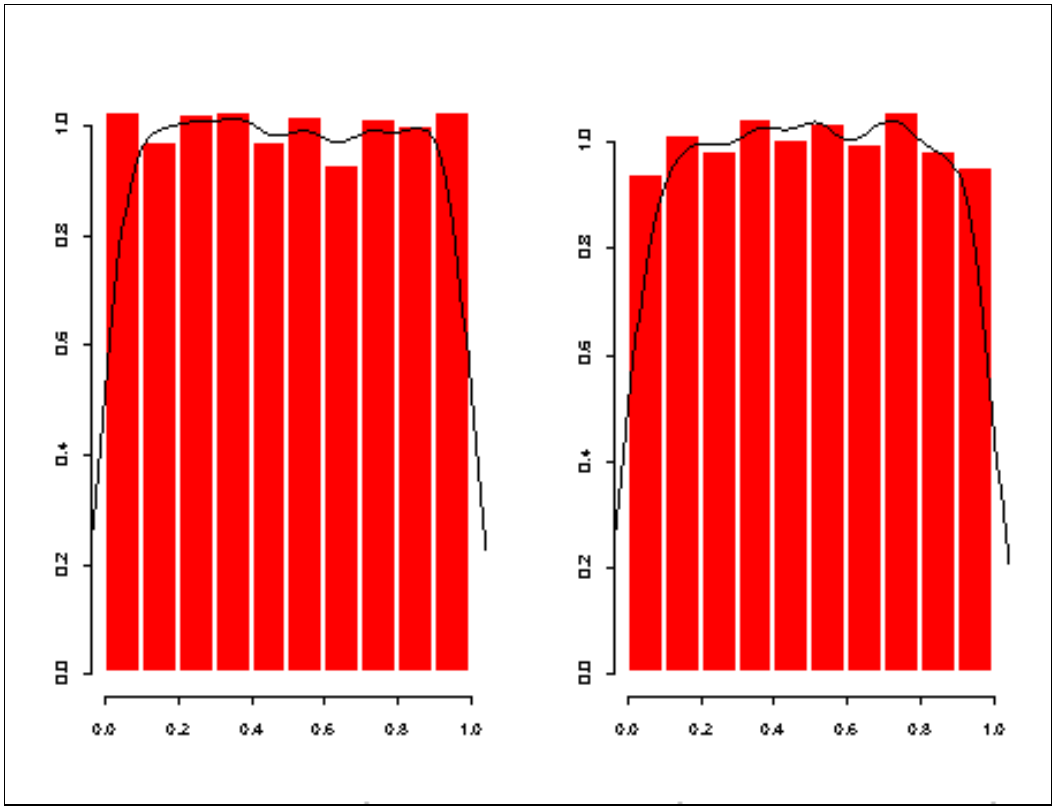
It is also necessary to choose *a*, *c*, M properly. The number M should be large. It may conveniently be taken as the computer's word size, since this makes the computation of X_i quite efficient. The choice of *a* is linked to the choice of M. If M is a power of 2, which is the case here, M is chosen equal to 2.E+32 and *a* is picked so that *a* mod8 be equal to 5 or *a*=5+8*k. Furthermore, *a* should be larger than $\sqrt{M}$, preferably larger than M/100, but smaller than $M - \sqrt{M}$. In general, *a* is taken equal to 3141592621. Finally, *c* should be an odd number and not a multiple of 5. In general, *c* is taken equal to 1. The random number generator will produce all different possible integer values of X_i in [0; (M-1)] (excepted the first value X_0) before starting to repeat. This choice of M, *a* and *c* ensures that the generator has a large periodicity and produces numbers that do not exhibit a strong dependency while following a uniform distribution.

If the user is interested in the generation of the numbers in [1; N], rather than in [0; (M-1)], the following formula will allow him to obtain them

$$[\ (\text{Int}\ (\text{dabs}\ (X_i\ /\ (M-1))*N))+1\ ] \qquad \textbf{(2)}$$

This formula **(2)** was implemented in FORTRAN language and is provided in Appendix E. The basic program generates numbers randomly with replacement. It allows the resamples calculation.

REMARK: no generator is able to generate a density truly uniform. S-PLUS may be used in order to check the reliability of our generator, given the seed number. A sample of 5,000 points from a uniform density distribution is created inside S-PLUS. Its probability distribution is compared to the probability distribution of the sample created with our generator. The results are shown in Figure 4.2.



With Random Generator & X_0=12 With S-PLUS Generator

Fig 4.2 Comparison between the Probability Distribution Generated with the FORTRAN Generator and the Probability Distribution Generated with the S-PLUS Generator

4.4-Conclusion

The nonparametric bootstrap method is a computationally intensive technique for making inferences about a population characteristic, γ. The evaluation is based on the estimator, Γ, related to the parameter, γ. A sample drawn from that population is used as starting point for the calculation. The nonparametric bootstrap method differs from the traditional parametric approach for inference in that it employs large numbers of resamples to estimate the shape of the Γ's sampling distribution. The calculated sampling distribution allows us to infer a confidence intervals around γ. Several different techniques were developed to build confidence intervals using the bootstrapped estimate of the sampling distribution. Among them, *the percentile method* was chosen to deal with the calculations presented in the next chapters. The method takes literally the fact that the bootstrap sampling distribution approximates the sampling distribution of Γ. The basic approach can be sum up as follows: an α-level confidence interval includes all the values of

Γ^* between the $\alpha/2$ and $(1-\alpha/2)$ percentiles of the bootstrap sampling distribution. That is, the endpoints of a 0.05 α-level confidence interval for Γ would be the values of Γ^* at the 2.5^{th} and 97.5^{th} percentile of the bootstrap sampling distribution.