

A Matlab Approach To Evaluate Product Quality

Hsin-Hung Wu

Department of Industrial Engineering and Management
Chien Kuo Institute of Technology
No. 1 Chea Sou North Road
Changhua City, Taiwan R.O.C. 500

Abstract

This study uses MATLAB as a programming tool and applies the bootstrap method to process capability analysis. The advantage of using MATLAB in bootstrap method is to make the bootstrap method much easier to implement and apply particularly in process capability analysis. An example is provided to further illustrate the easy use of MATLAB in bootstrap method.

Keywords : Process capability analysis, Process capability index, Bootstrap method, Data Analysis

1. Introduction

The bootstrap method was first introduced in 1979 by Bradley Efron and has become one of the important techniques in data analysis, such as in statistical process control areas [4,6,14,16]. The advantage of this technique is that no underlying assumptions are required before researchers conduct any analysis. In a statistical context, the bootstrap method describes a way to generate an entire distribution of a population starting from only a sample.

The bootstrap method is a non-parametric but computer intensive method for making probability-based inferences about population parameters without theoretical assumptions. Because of its complexity and a lack of understanding, the bootstrap method has been limited to those who are particularly interested in mathematics and statistics. For those who are interested in bootstrap method without strong background in mathematics, statistics, and programming abilities, the implementation is somewhat difficult.

Therefore, Woodroof [15] has developed a template using Lotus 1-2-3 spreadsheet to implement this technique for those who are interested in business research can find an effective way to learn it.

MATLAB, on the other hand, is one of the very popular programming languages for electrical, electronic, and mechanical engineering students to do simulations in many areas. In addition, industrial engineering and management students have become familiar with MATLAB because this computer language is not as complicated as those of FORTRAN or others. In fact, simple calculations and simulations can be conducted by MATLAB with very short programs. It is worth noting that MATLAB provides the statistical toolbox to further simplify the calculations of basic statistics and uses vectors or matrices in calculations and displaying results.

The purpose of this study is to use the built-in bootstrap functions in MATLAB as an approach to evaluate product quality particularly when the process data are non-normally distributed. The percentile-based process capability indices (PCIs) are used along with the bootstrap method, which does not require any assumption on data analysis, to provide the further information for each PCI. When the iteration of the re-sampling process is 1000 times, it is believed that the true values of product quality can be

analyzed between minimum and maximum values or by confidence intervals [5,6,14]. A more conservative approach in evaluating product quality can be provided based upon the minimum value of 1000 iterations. That is, the product quality would not be worse than the minimum value of 1000 iterations.

This paper is organized as follows: Sections 2, 3, and 4 describe the bootstrap method, the popularity of MATLAB software and percentile-based PCIs, respectively. The research method, an example, and conclusions are summarized in Sections 5, 6, and 7, respectively.

2. The Bootstrap Method

Wasserman and Franklin [14] described the bootstrap method as follows: Let a sample of size n be taken from a process that has distribution F .

$$x_1, x_2, \dots, x_n \sim F. \quad (1)$$

A bootstrap sample is one of size n drawn with replacement from the original sample and is denoted as

$$x_1^*, x_2^*, \dots, x_n^*. \quad (2)$$

There are a total of n^n such possible

samples. If the bootstrap is applied to the C_{pk} index, these samples are used to compute n^n values of \hat{C}_{pk}^* (not all unique); each of which would estimate \hat{C}_{pk} , and their entire collection would constitute the bootstrap distribution for \hat{C}_{pk} . In fact, this bootstrap distribution is dependent upon the original sample drawn from the process.

This technique is to put probability mass $1/n$ on each original x_i to create an empirical probability distribution function of the data, \hat{F} . This \hat{F} is the non-parametric maximum likelihood estimate of F and will converge to F as $n \rightarrow \infty$. The bootstrap sampling is equivalent to sampling from \hat{F} with replacement. The bootstrap distribution of \hat{C}_{pk}^* is an estimate of the distribution of \hat{C}_{pk} .

In actual practice, for small samples, calculations of the complete bootstrap distribution are virtually impossible. If $n = 10$, there are 10^{10} values of \hat{C}_{pk}^* to compute by drawing all the possible samples. Therefore, empirical work shows that a minimum of 1000 bootstrap samples should be enough to compute confidence interval estimates [5,6,14].

Three major confidence intervals were

seen in literature [7], including the standard bootstrap, the percentile bootstrap, and the biased-corrected percentile bootstrap methods. For the standard bootstrap method, if notations of \hat{C} and $\hat{C}^*(i)$ are denoted as the estimator of a capability index and the associated ordered bootstrap estimates, the sample average of \hat{C}^* and the sample standard deviation of S_C^* from 1000 bootstrap estimates of $\hat{C}^*(i)$ are presented as

$$\hat{C}^* = \frac{1}{1000} \sum_{i=1}^{1000} \hat{C}^*(i) \quad (3)$$

and

$$S_C^* = \sqrt{\frac{1}{999} \sum_{i=1}^{1000} (\hat{C}^*(i) - \hat{C}^*)^2} \quad (4)$$

The quantity S_C^* is an estimator of the standard deviation of \hat{C} . The $(1 - 2\alpha)$ 100% confidence interval of C is

$$\hat{C} \pm z_\alpha S_C^* \quad (5)$$

where z_α is the upper α quantile of the standard normal distribution.

For the percentile bootstrap method, if the ordered collection of $\hat{C}^*(i)$, the α percentage, the $(1 - \alpha)$ percentage point, and 1000 iterations are used, the confidence interval of

$(1 - 2\alpha)$ 100% for C is

$$[\hat{C}^*(1000\alpha), \hat{C}^*(1000(1 - \alpha))]. \quad (6)$$

If a 90% confidence interval is used, Equation (6) becomes $[\hat{C}^*(50), \hat{C}^*(950)]$.

It is possible that the bootstrap distributions obtained using a sample of the complete bootstrap distribution may be shifted higher or lower than expected. Thus, the biased-corrected percentile bootstrap method was developed to correct this potential bias. The procedures are as follows: First, using the ordered distribution of \hat{C}^* to calculate the probability of P_0 :

$$P_0 = P_r[\hat{C}^* \leq \hat{c}] \quad (7)$$

Second, calculate z_0 , P_L , and P_U , the formulas are

$$z_0 = \Phi^{-1}(P_0), \quad (8)$$

$$P_L = \Phi(2z_0 - z_\alpha), \quad (9)$$

and

$$P_U = \Phi(2z_0 + z_\alpha), \quad (10)$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. Finally, if the iteration is 1000 times, the confidence

interval is

$$[\hat{C}^*(1000P_L), \hat{C}^*(1000P_U)]. \quad (11)$$

3. Matlab Software

Schilling and Harris [13] believed that the key to the successful application of numerical methods is effective software, which includes the assumptions that students are familiar with the fundamentals of MATLAB programming environment. For MATLAB users, a numerical toolbox of MATLAB functions is available. This toolbox includes a set of main-program support functions, which are low-level utility functions designed to ease user interaction and the display of numerical results. In addition, in Version 5.30 (R11), the functions in toolbox include a lot of very popular statistical functions in business and/or industrial engineering and management areas, such as probability density functions, cumulative distribution functions, random number generators, linear and non-linear models, statistical process control, and descriptive statistics with mean, median, standard deviation, skewness, kurtosis, and to name a few.

Borse [1] described MATLAB as the effective and efficient use of the powerful numerical analysis package that relies on

programmer's ability to construct, manipulate, and solve matrix equations. In MATLAB software, the function of the bootstrap method was mainly based upon the algorithm provided by Efron and Tibshirani [5]. The advantages of MATLAB are summarized as follows [1]:

1. Most computer programs are heavily dependent upon a variety of "looping" procedure to execute summations or to implement iterative techniques. A single program will contain essentially identical looping structures tens of times, greatly complicating the writing of the code.
2. The majority of the problems repeatedly call on a relatively small set of "subfunctions," usually the same set for a variety of problems. The MATLAB has its encyclopedic collection of subprograms, called M-files, for the solution of nearly any numerical problems.
3. The MATLAB provides user-friendly and easy-to-use graphics capabilities, particularly in two- or three-dimensional graphs.

Moreover, users can develop their own M-files to solve particular numerical problems. Because this computer language is not as complicated as those of FORTRAN or others, more and more students have begun to learn MATLAB as a programming tool. In addition, simple calculations, basic statistics, and complicated simulations can be conducted by MATLAB with very short

programs along with a sufficient numerical toolbox.

For the bootstrap method, statistical toolbox of the MATLAB program provides a "bootstrp" code for bootstrap statistics, which can be found in `toolbox\stats\Descriptive Statistics` from Help Window. In this paper, an example is provided to illustrate the use of MATLAB in bootstrapping.

4. Percentile-Based Process Capability Indices

Pyzdek [12] has stated that normal distributions are not the norm based upon his industrial experience. The normality-based C_p , C_{pk} , C_{pm} , and C_{pmk} indices cannot be applied when the underlying distribution belongs to non-normal distributions. One of the approaches to deal with non-normal data recommended by Kotz and Johnson [8] is to develop non-normality-based process capability indices.

Several non-normality-based process capability indices have been proposed, such as the weighted variance approaches, Johnson-Kotz-Pearn approach, the Johnson system-based method, percentile-based process capability indices, and to name a few [2,8,9,11,17,18,19,20,22,23]. The most commonly used PCIs are percentile-based

PCIs which can be applied to both normal and non-normal data [2,11].

Clements [2] used percentile values to construct the C_p and C_{pk} indices for non-normal populations. In addition, Pearn and Kotz [11] have adopted the similar philosophy to develop the C_{pm} and C_{pmk} indices. The formulas are described as follows:

$$C_p = \frac{USL - LSL}{U_p - L_p}, \tag{12}$$

$$C_{pk} = \min\left(\frac{USL - M}{U_p - M}, \frac{M - LSL}{M - L_p}\right), \tag{13}$$

$$C_{pm} = \frac{USL - LSL}{6\sqrt{\left(\frac{U_p - L_p}{6}\right)^2 + (M - T)^2}}, \tag{14}$$

and

$$C_{pmk} = \min\left(\frac{USL - M}{3\sqrt{\left(\frac{U_p - M}{3}\right)^2 + (M - T)^2}}, \frac{M - LSL}{3\sqrt{\left(\frac{M - L_p}{3}\right)^2 + (M - T)^2}}\right), \tag{15}$$

where USL and LSL are upper specification and lower specification limits, U_p , M , and L_p are the 99.865,50 (median), and 0.135 percentiles, respectively, and T is the target value. $U_p - L_p$ is used to represent 6σ , the width of the central $\pm 3\sigma$ interval

of the normal distribution, containing 99.73 percent of the population.

The values of U_p , M , and L_p can be determined by two approaches. Clements [2] has provided mathematical equations along with Tables 1a, 1b, and 2 to compute U_p , M , and L_p . The capability index values using Clements' tables will be provided in Section 6. The other approach is to calculate U_p , M , and L_p directly from the sample. If the sample size is not large enough, U_p and L_p cannot be directly estimated. Dudewicz and Mishra [3] have provided the formulas to estimate percentile values for different sample sizes. If the sample of size is 100, the values of U_p and L_p are substituted by the maximum and minimum values, which will be used in the bootstrap method in Section 6. For further information about percentile-based PCIs, please refer to Wu [21].

5. Research Method

The purpose of this study is to use MATLAB as a programming language to deal with a non-normal data set with a sample of size 100 by applying the bootstrap method. The percentile-based C_{pm} and C_{pmk} indices are used for process

capability analysis. By applying the bootstrap method, the minimum, average, and maximum values for each PCI can be further examined. In addition, the confidence intervals of the C_{pm} and C_{pmk} indices will be provided in Section 6. The decisions thus can be made based upon those statistics.

6. An Example

The raw data of 100 observations borrowed from Wu [23] were recorded in Table 1, and the histogram is provided in Figure 1. According to the normality test conducted by Minitab [10], since the P-values are less than 0.01, the data may come from a non-normal distribution. The USL , T , and LSL were set to 573.60, 573.50, and 573.40 millimeters (mm), respectively. There is no any observation outside the specification limits. In addition, only three

observations, 573.42, 573.56, and 573.57, are close to the specification limits, and most of the observations are located at 573.51 mm shown in Figure 1. The sample average and sample standard deviation are 573.498 and 0.019, respectively, with skewness of -0.27 and kurtosis of 8.12.

If the normality-based C_{pm} and C_{pmk} indices are used, the values are 1.74 and 1.71, respectively. Since the raw data might belong to non-normal populations, Equations (14) and (15) should be used instead of the normality-based PCIs. The maximum, median, and minimum values of this data set are 573.57, 573.50, and 573.42, respectively, and the respective values of U_p , M , and L_p are 573.57, 573.50, and 573.42. Based upon Equations (14) and (15), the C_{pm} and C_{pmk} values are 1.33 and 1.25, respectively. If Clements' tables are used, the values of U_p ,

Table 1 The Raw Data from A Bicycle Wheel Manufacturing Company (Source: Wu [23])

573.51	573.50	573.52	573.52	573.51	573.50	573.49	573.50	573.50	573.51
573.50	573.51	573.42	573.51	573.50	573.46	573.50	573.52	573.47	573.50
573.49	573.48	573.50	573.49	573.50	573.49	573.52	573.51	573.51	573.49
573.44	573.49	573.50	573.50	573.47	573.50	573.50	573.48	573.51	573.47
573.53	573.56	573.57	573.50	573.50	573.51	573.49	573.50	573.49	573.50
573.50	573.49	573.48	573.48	573.48	573.49	573.50	573.47	573.51	573.50
573.50	573.50	573.50	573.50	573.50	573.52	573.51	573.50	573.50	573.48
573.49	573.50	573.50	573.51	573.49	573.50	573.50	573.50	573.50	573.50
573.47	573.52	573.51	573.50	573.50	573.48	573.50	573.49	573.49	573.50
573.52	573.50	573.50	573.49	573.51	573.50	573.46	573.50	573.50	573.51

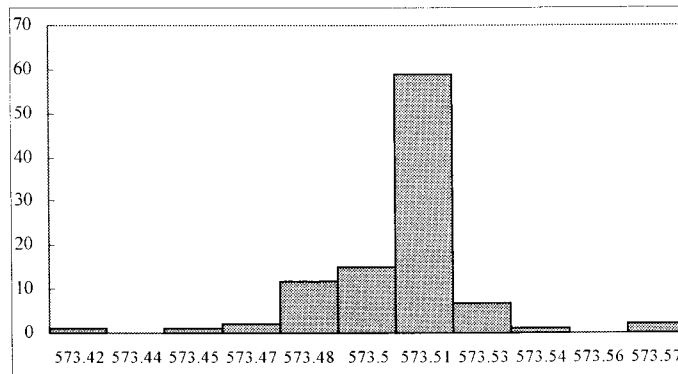


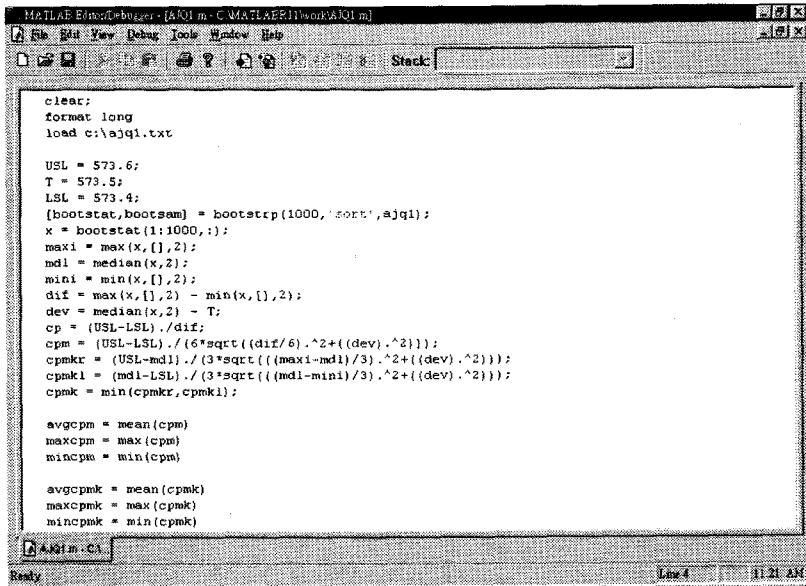
Figure 1. The Histogram of the 100 Observations in Table 1

M , and L_p are 573.5629, 573.4984, and 573.4135, respectively. The C_{pm} and C_{pmk} values using Equations (14) and (15) are 1.34 and 1.16.

Another approach to deal with non-normal process data is to find the true distribution of this raw data set, and the Johnson system is one of the effective families of distributions to fit the data set [18,20]. If the true distribution of the data set is known, further discussions can be made. However, not every type of process data can be fitted well by any one of the Johnson system. Therefore, it might be wise to use the bootstrap method to further investigate the properties of a data set without making any assumptions in advance.

The MATLAB code for this example is provided in Figure 2, where `ajql.txt` is the file name to store the 100 observations shown in Table 1. The data set can be

represented by either a 100×1 or 1×100 vector. In this example, this 100 observations were stored as a 100×1 vector. In Figure 2, the terms of “median”, “max”, “min”, and “mean” are built-in functions. The results are summarized in Figure 3, where “avgcpm”, “maxcpm”, and “mincpm” represent the average, maximum, and minimum values of the C_{pm} index, respectively. Based upon the simulation, the maximum, average, and minimum values of the C_{pm} index are 4.00, 1.58, and 1.33, respectively. The results generated by MATLAB were less than ten seconds by a personal computer with PII 400 processor. For the C_{pmk} index, the maximum, average, and minimum values are 3.33, 1.37, and 1.25, respectively. Clearly, the product quality would not be worse than 1.33 and 1.25 based upon the simulated results.



```

clear;
format long
load c:\ajql.txt

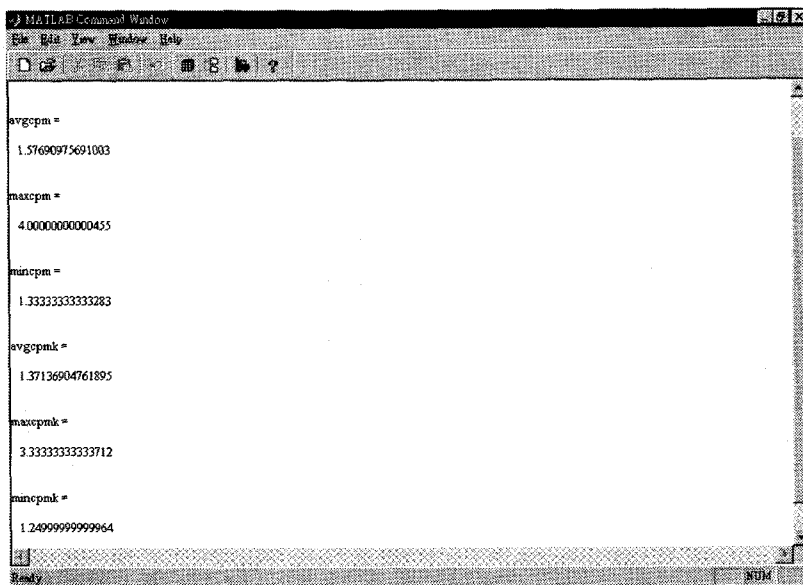
USL = 573.6;
T = 573.5;
LSL = 573.4;
[bootstat,bootsum] = bootstrp(1000,'sort',ajql);
x = bootstat(1:1000,:);
max1 = max(x,[],2);
md1 = median(x,2);
min1 = min(x,[],2);
dif = max(x,[],2) - min(x,[],2);
dev = median(x,2) - T;
cp = (USL-LSL)./dif;
cpm = (USL-LSL)./(6*sqrt((dif/6).^2+((dev).^2)));
cpmk1 = (USL-md1)./(3*sqrt(((max1-md1)/3).^2+((dev).^2)));
cpmk1 = (md1-LSL)./(3*sqrt(((md1-min1)/3).^2+((dev).^2)));
cpmk = min(cpmk1,cpmk1);

avgcpm = mean(cpm)
maxcpm = max(cpm)
mincpm = min(cpm)

avgcpmk = mean(cpmk)
maxcpmk = max(cpmk)
mincpmk = min(cpmk)

```

Figure 2 The MATLAB Code for the Example



```

avgcpm =
1.57690975691003

maxcpm =
4.0000000000455

mincpm =
1.3333333333283

avgcpmk =
1.37136904761895

maxcpmk =
3.3333333333712

mincpmk =
1.2499999999964

```

Figure 3 The Simulated Results for the Example

If confidence intervals of the C_{pm} and C_{pmk} indices are concerned, Equations (5), (6), and (11) can be used. For instance, if the percentile bootstrap method is applied and the iteration is 1000 times, the 95 and 99 percent confidence intervals of the C_{pm} and C_{pmk} indices are summarized in Table 2. It is interesting to note that the lower confidence bounds of the C_{pm} index for 95% and 99% are identical, which are equivalent to the minimum value of the C_{pm} index. In addition, the lower confidence bounds of the C_{pmk} index for 95% and 99% are equivalent, which are equivalent to the minimum value of the C_{pmk} index. Based upon Table 2, if the 95% or 99% of the confidence intervals of the C_{pm} and C_{pmk} indices are used, the product quality is not going to be lower than 1.33 and 1.25, respectively.

If the target value is switched from 573.50 to 573.52, the C_{pm} and C_{pmk} values using Equations (14) and (15) are 1.04 and 1.00, respectively. In addition, using Clements' tables along with Equations (14) and (15) would result in 1.01 and 0.92 for the C_{pm} and C_{pmk} indices. On the other hand, if the bootstrap method is used, the minimum values of the C_{pm} and C_{pmk} indices after 1000 re-sampling process are 1.04 and 1.00, respectively. The average values of the C_{pm} and C_{pmk} indices are 1.12 and 1.04. Clearly, using the bootstrap method in process capability analysis provides much more information than the point estimates computed from the data set. Further discussions and results can be simulated by adding up appropriate statistical terms in MATLAB programs.

7. Conclusions

The bootstrap method is a non-parametric but computer intensive method for making

Table 2 The 95 and 99 Percent Confidence Intervals of the C_{pm} and C_{pmk} Indices

	C_{pm}	C_{pmk}
95% Upper Confidence Bound	2.2222	1.6667
95% Lower Confidence Bound	1.3333	1.2500
99% Upper Confidence Bound	3.3333	2.5000
99% Lower Confidence Bound	1.3333	1.2500

probability-based inference about parameters without theoretical assumptions. For those who are not familiar with statistics and do not have strong background in programming ability, MATLAB provides a "short cut" to use the bootstrap method in process capability analysis. The example is also demonstrated to show the convenience of using MATLAB in bootstrapping. It is believed that the bootstrap method would become much easier to apply in other industrial engineering related research topics if the MATLAB software is used as a programming tool.

References

1. Borse, G.J., *Numerical Methods with MATLAB: A Resource for Scientists and Engineers*, PWS Publishing Company, Boston, 1997.
2. Clements, J.A., "Process Capability Calculations for Non-Normal Distributions," *Quality Progress*, 22(9), 95-100 (1989).
3. Dudewicz, E.J. and S.N. Mishra, *Modern Mathematical Statistics*, John Wiley & Sons, 1988.
4. Efron, B., "Bootstrap Methods: Aother Look at the Jackknife," *Annals of Statistics*, 7, 1-6 (1979).
5. Efron, B. and R.J. Tibshirani, *An Introduction to the Bootstrap*, Chapman & Hall, New York, 1993.
6. Franklin, L.A. and G. Wasserman, "Standard Bootstrap Confidence Interval Estimates of C_{pk} ," *Computers and Industrial Engineering*, 21, 129-133 (1991).
7. Franklin, L.A. and G.S. Wasserman, "Bootstrap Lower Confidence Limits for Capability Indices," *Journal of Quality Technology*, 24(4), 196-202 (1992).
8. Kotz, S. and N.L. Johnson, *Process Capability Indices*. Chapman & Hall, (1993).
9. Kotz, S. and C.R. Lovelace, *Process Capability Indices in Theory and Practice*. Arnold, (1998).
10. Minitab, *Minitab User's Guide 2: Data Analysis and Quality Tools, Release 13*, Minitab, Inc., 2000.
11. Pearn, W.L. and S. Kotz, "Application of Clements' Method for Calculating Second- and Third-Generation Process Capability Indices for Non-Normal Pearsonian Populations," *Quality Engineering*, 7, 139- 145 (1994-95).
12. Pyzdek, T., "Why Normal Distributions Aren't [All That Normal]," *Quality Engineering*, 7(4), 769-777 (1995).
13. Schilling, R.J. and S.L. Harris, *Applied Numerical Methods for Engineers Using MATLAB and C*, Brooks/Cole Publishing Company, 1999.
14. Wasserman, G.S. and L.A. Franklin, "Standard Bootstrap Confidence Interval

- Estimates of C_{pk} ," *Computers and Industrial Engineering*, 22(2), 171-176 (1992).
15. Woodroof, J., "Bootstrapping: as easy as 1-2-3," *Journal of Applied Statistics*, 27(4), 509-517 (2000).
 16. Wu, Z. and Q. Wang, "Bootstrap Control Charts," *Quality Engineering*, 9(1), 143-150 (1996-97).
 17. Wu, H.-H., Swain, J.J., Farrington, P.A., and S.L. Messimer, "A Weighted Variance Capability Index for General Non-Normal Processes," *Quality and Reliability Engineering International*, 15(5), 397-402 (1999).
 18. Wu, H.-H. and T.L. Liu, "An Integrated Approach of Process Capability Analysis to Improve Product Quality," *Proceedings of the Sixth International Conference on Automation Technology*, Vol. 2, 965-972 (2000).
 19. Wu, H.-H., "The Performance of A New Process Capability Index for Skewed Process Data," *Journal of Quality*, 7(1), 97-114 (2000).
 20. Wu, H.-H. and S.K. Fan, "Process Capability Analysis for both Normal and Non-Normal Data," *Journal of Quality*, 7(2), 93-115 (2000).
 21. Wu, H.-H., "Performance Problems of Families of Non-Normal Process Capability Indices," *Quality Engineering*, 13(3), 383-388 (2001).
 22. Wu, H.-H. and J.J. Swain, "A Monte Carlo Comparison of Capability Indices when Processes are Non-Normally Distributed," *Quality and Reliability Engineering International*, 17(3), 219-231 (2001).
 23. Wu, H.-H., "Process Capability Indices for Skewed Process Data," *International Journal of Industrial Engineering - Theory, Applications and Practice*, 8(3), 210-219 (2001).
-