利用拔靴法於數據分析：使用 MATLAB 方法

Bootstrapping in dara analysis：A matlab approach

吳信宏[1]　郭廣洋[2]

摘要

拔靴法為無母數且為電腦密集的方法，採用機率的推論預估母體參數，不需要對母體有任何理論假設。由於拔靴法相當的複雜且不易了解，因此並未廣泛的運用於社會科學領域。雖然拔靴法可以透過使用 Lotus 1-2-3 計算，但是在效能上面仍然不盡理想。本文的目的在於介紹與解釋如何使用 MATLAB 電腦語言簡化拔靴法的使用。

關鍵詞：MATLAB、拔靴、資料分析、無母數

Abstract

The bootstrap is a non-parametric but computer intensive method for making probability-based inferences about population parameters without theoretical assumptions. However, due to its complexity and a lack of understanding, the bootstrap has not been widely used in social science fields. Using Lotus 1-2-3 in bootstrapping is helpful but still somewhat ineffective. The purpose of this paper is to introduce and explain how the bootstrap method can be applied much easier using MATLAB computer language.

***Keywords*: MATLAB, Bootstrap, Data analysis, Non-parametric**

1. Introduction

The bootstrap method first introduced in 1979 by Bradly Efron has become one of the important techniques in data analysis (Efron, 1979; Franklin and Wasserman, 1991; Wu and Wang, 1996-97; Wu, 2001). It is a non-parametric but computer-intensive method for making probability-based inferences about population parameters without theoretical assumptions. In a statistical context, the bootstrap method describes a way to generate an entire distribution of a population starting from only a sample.

Because of its complexity and a lack of understanding, the bootstrap method has not been widely used in social science fields (Woodroof, 2000). Two major reasons are summarized as follows (Woodroof, 2000): (1) There is a lack of understanding of the

[1]臺中健康暨管理學院資訊管理學系副教授
[2]中央警察大學交通學系副教授

bootstrap technique, and (2) bootstrapping has traditionally been difficult to implement. Therefore, Woodroof (2000) has developed a template using Lotus 1-2-3 spreadsheet to effectively find a way to learn and implement the bootstrap method in business research.

Using Lotus 1-2-3 in bootstrapping is helpful, however, it is still somewhat ineffective. As more and more management students are familiar with MATLAB computer language, this powerful computer language can be used in data analysis, especially in the bootstrap method. The purpose of this paper is to introduce and explain how the bootstrap method can be applied much easier using MATLAB computer language. By developing a template using MATLAB, this paper provides those who are interested in bootstrap method without strong backgrounds in programming but with an effective way to learn and implement it with some easy MATLAB codes.

2. The Matlab Tool

Schilling and Harris (1999) believed that the key to the successful application of numerical methods is effective software, which includes the assumptions that the 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 Version 5.30 (R11), the functions in toolbox includes various popular statistical functions in business and/or industrial engineering and management areas, including probability density function, random number generators, linear and non-linear models, descriptive statistics with mean, median, standard deviation, skewness, and kurtosis, and to name a few.

Borse (1997) 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 (1993). The advantages of MATLAB are summarized as follows:

1. 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 various 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.

Currently, MATLAB is one of the very popular programming languages for Electrical, Electronic, and Mechanical Engineering students to do simulations in many areas. More importantly, some management-oriented students have become more and more familiar with

MATLAB because this computer language is not as complicated as those of FORTRAN or others. The MATLAB software provides the statistical toolbox to further simplify the calculations of basic statistics. It is worth to note that MATLAB uses vectors or matrices in calculations, and, of course, the results are displayed as vectors or matrices as well.

The 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, two examples are demonstrated to illustrate the use of MATLAB in bootstrapping.

3. Examples

The data set used in the first example is from Woodroof (2000) and summarized in Table 1. The data are the average LSAT scores and the average undergraduate GPAs from 15 classes entering law school. The interest is the Pearson correlation coefficients with the original data and the 8000 iterations by the bootstrap method. Figure 1 provides the MATLAB program, and the MATLAB code of “corrcoef” is a built-in function to calculate the value of correlation coefficient between LSAT and GPA. Both LSAT and GPA are presented as row vectors.

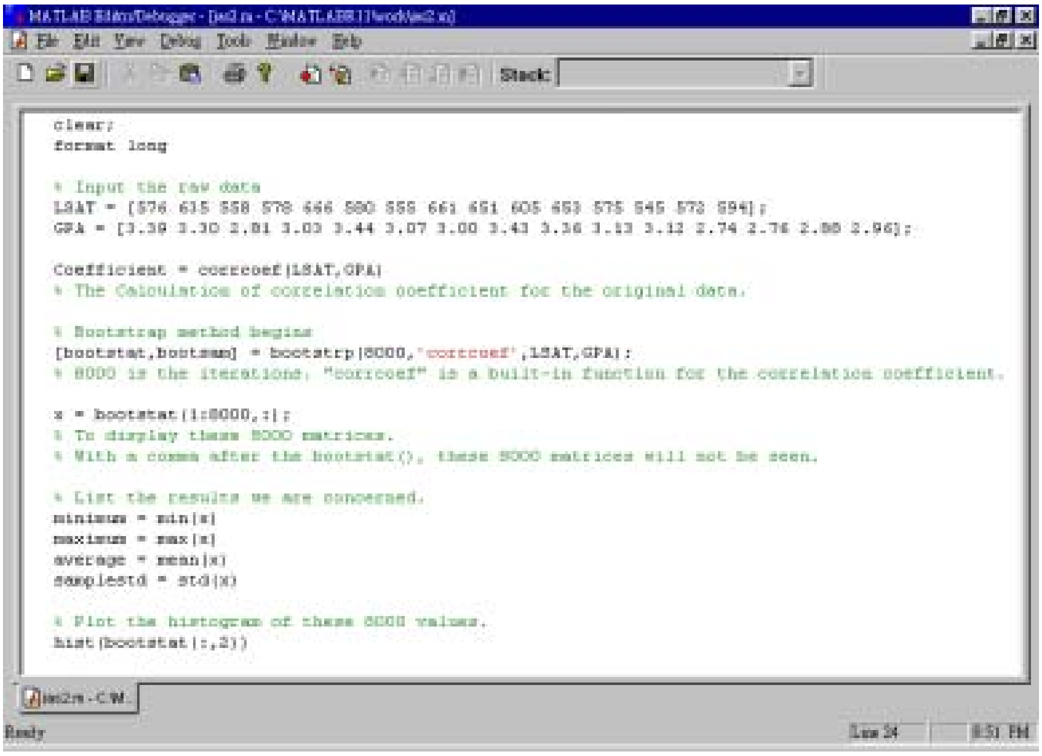
Table 1 The Original Data from Woodroof (2000)

	LSAT	GPA		LSAT	GPA		LSAT	GPA
1	576	3.39	6	580	3.07	11	653	3.12
2	635	3.30	7	555	3.00	12	575	2.74
3	558	2.81	8	661	3.43	13	545	2.76
4	578	3.03	9	651	3.36	14	572	2.88
5	666	3.44	10	605	3.13	15	594	2.96

The command of “clear” typically indicates the beginning of a new program, and “format long” represents the displayed numbers with 15 digits, while “format short” can only provide 5 digits in accuracy. On line 6, the command is to compute the Pearson correlation coefficient based upon the original data. The code of “bootstrp(8000,’corrcoef’,LSAT,GPA)” on line 9 is to resample 15 pairs of LSAT and GPA values for 8000 times, and the 8000 correlation coefficients can be computed by the built-in “corrcoef” function as well. The code of “bootstat(1:8000,:)” is to list these 8000 Pearson correlation coefficients. If a comma is added, then these 8000 values will not be seen in MATLAB Command Window.

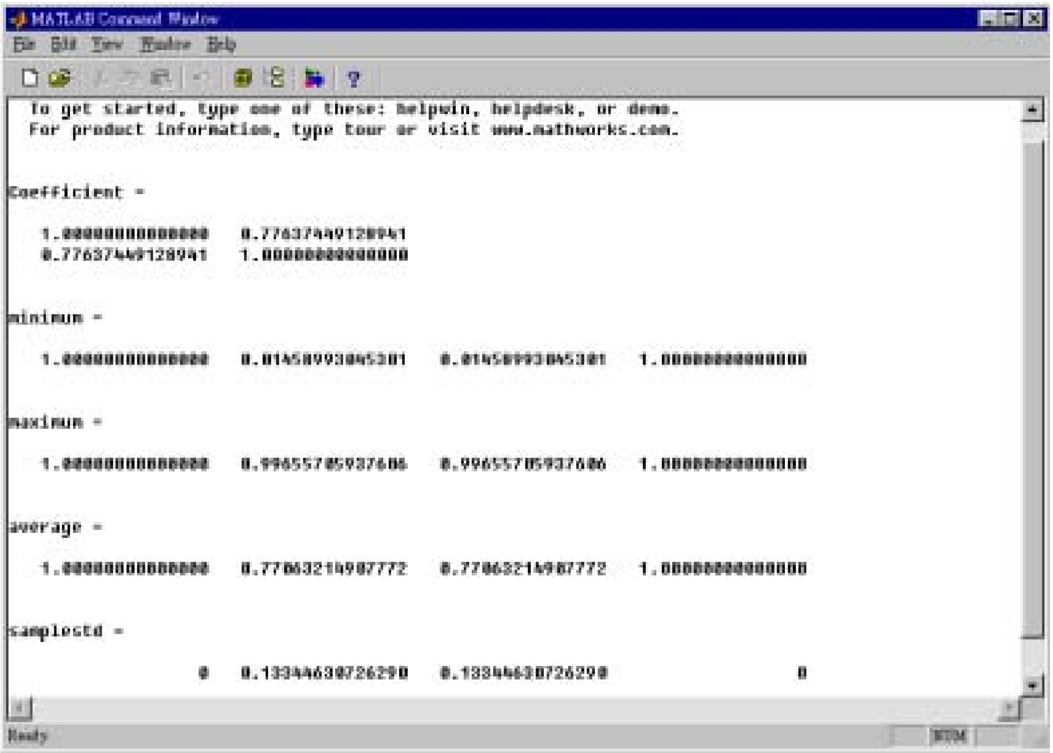
The statistics of the 8000 Pearson correlation coefficients we are concerned include the maximum value, minimum value, sample average, and sample standard deviation. In MATLAB program, built-in functions can be found to represent the maximum value, minimum value, sample average, and sample standard deviation by max(x), min(x), mean(x), and std(x), respectively. Moreover, The histogram can be plotted using “hist(bootstat(:,2))” for the 8000 Pearson correlation coefficients. To run the program, go to “tools” and then select “run” of MATLAB Editor/Debugger so that the results will be displayed in Command

Window. For 8000 iterations, it only takes less than 20 seconds to generate the results by PC. Finally, the results and histogram are concluded in Figure 2 and 3, respectively.


MATLAB Editor/Debugger - [ae2.m - C:\MATLABR11\work\ae2.m]
File Edit View Debug Tools Window Help
Stack:
ae2.m - C:\W...
Ready
Line 24
8:51 PM

```
clear;
format long

% Input the raw data
LSAT = [576 635 558 578 666 580 555 661 651 605 653 575 545 572 594];
GPA = [3.39 3.30 2.81 3.03 3.44 3.07 3.00 3.43 3.36 3.13 3.12 2.74 2.76 2.88 2.96];

Coefficient = corrcoef(LSAT,GPA)
% The Calculation of correlation coefficient for the original data.

% Bootstrap method begins
[bootstat,bootsam] = bootstrp(8000,'corrcoef',LSAT,GPA);
% 8000 is the iterations; "corrcoef" is a built-in function for the correlation coefficient.

x = bootstat(1:8000,:);
% To display these 8000 matrices.
% With a comma after the bootstat(), these 8000 matrices will not be seen.

% List the results we are concerned.
minimum = min(x)
maximum = max(x)
average = mean(x)
samplestd = std(x)

% Plot the histogram of these 8000 values.
hist(bootstat(:,2))
```

Figure 1 The MATLAB Program of Example 1



```
  To get started, type one of these: helpwin, helpdesk, or demo.
  For product information, type tour or visit www.mathworks.com.


Coefficient =

   1.00000000000000   0.77637449128941
   0.77637449128941   1.00000000000000


minimum =

   1.00000000000000   0.01458993045301   0.01458993045301   1.00000000000000


maximum =

   1.00000000000000   0.99655705937606   0.99655705937606   1.00000000000000


average =

   1.00000000000000   0.77063214987772   0.77063214987772   1.00000000000000


samplestd =

                  0   0.13344630726290   0.13344630726290                  0
```

Figure 2 The Results Generated by MATLAB Program of Example 1

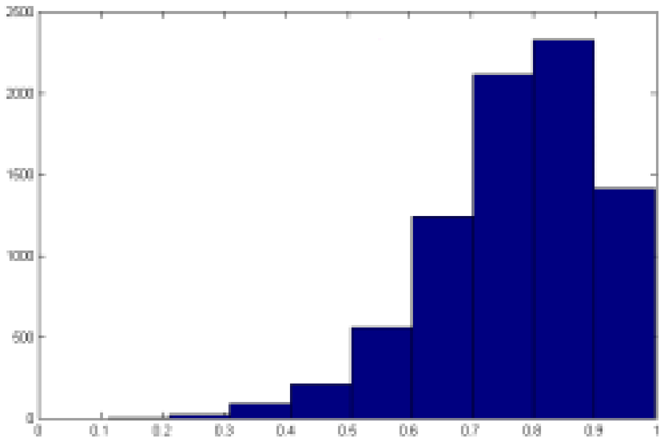


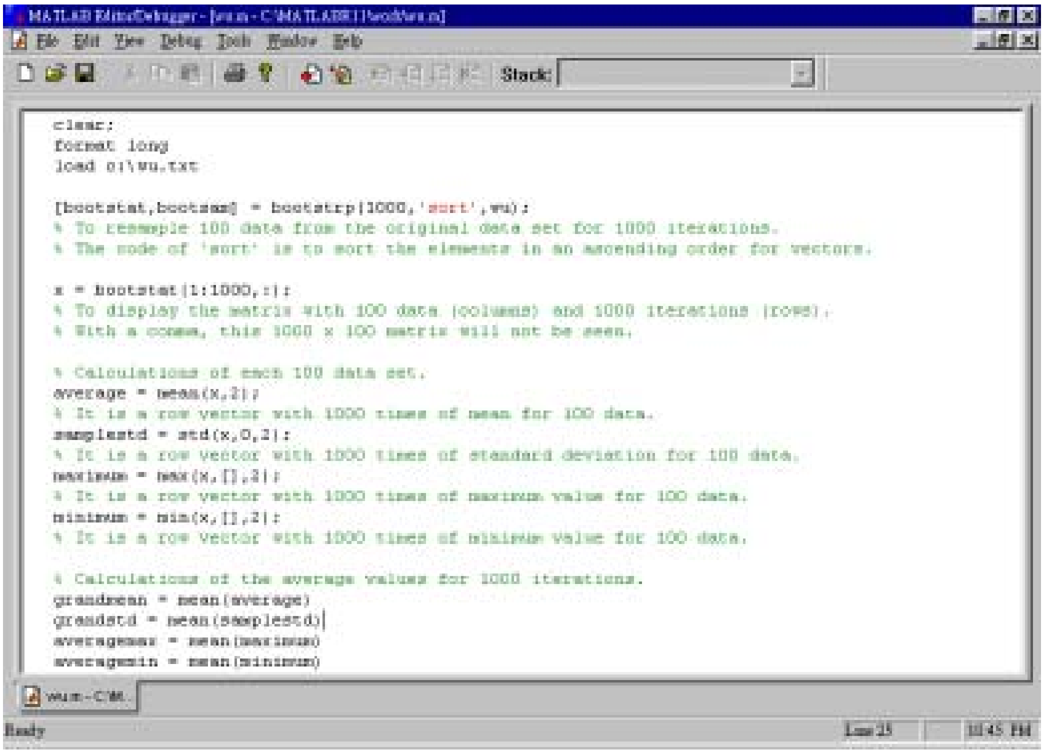
Figure 3 The Histogram of the 8000 Correlation Coefficients

In Figure 2, five matrices for the raw data, maximum, minimum, mean, and standard deviation are all 2 x 2. The results are concluded as follows: The Pearson correlation coefficient in Figure 1 is 0.77637. For 8000 iterations, the maximum value, minimum value, average, and standard deviation of the Pearson correlation coefficients are 0.99656, 0.01459, 0.77063, and 0.13345, respectively. Compared with the results provided in Figure 4 (Page 514) by Woodroof (2000), the Pearson correlation coefficient in Figure 1 is exactly the same, while the maximum value, minimum value, average, and standard deviation are very close. However, using MATLAB in this example is much easier, and the results can be displayed much quicker.

Table 2 The Data Set Borrowed from Wu (2000)

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

The second example is to resample the data, borrowed from Wu (2000) and summarized in Table 2, for 1000 iterations. The original data set has a sample mean of 573.50, a standard deviation of 0.018872, and the largest and smallest values are 573.57 and 573.42, respectively. If the statistics we are concerned include the grand mean, grand standard deviation, average maximum value, and average minimum value of the 1000 iterations, the MATLAB program is provided in Figure 4. In Figure 4, the raw data in Table 2 was stored as a column vector, entitled wu.txt. The code of “sort” is to sort the data in an ascending order for vectors.


MATLAB Editor/Debugger - [wu.m - C:\MATLABR11\work\wu.m]
File Edit View Debug Tools Window Help
Stack:
wu.m - C:\M...
Ready
Line 25
10:45 PM

```
clear;
format long
load c:\wu.txt

[bootstat,bootsam] = bootstrp(1000,'sort',wu);
% To resample 100 data from the original data set for 1000 iterations.
% The code of 'sort' is to sort the elements in an ascending order for vectors.

x = bootstat(1:1000,:);
% To display the matrix with 100 data (columns) and 1000 iterations (rows).
% With a comma, this 1000 x 100 matrix will not be seen.

% Calculations of each 100 data set.
average = mean(x,2);
% It is a row vector with 1000 times of mean for 100 data.
samplestd = std(x,0,2);
% It is a row vector with 1000 times of standard deviation for 100 data.
maximum = max(x,[],2);
% It is a row vector with 1000 times of maximum value for 100 data.
minimum = min(x,[],2);
% It is a row vector with 1000 times of minimum value for 100 data.

% Calculations of the average values for 1000 iterations.
grandmean = mean(average)
grandstd = mean(samplestd)
averagemax = mean(maximum)
averagemin = mean(minimum)
```

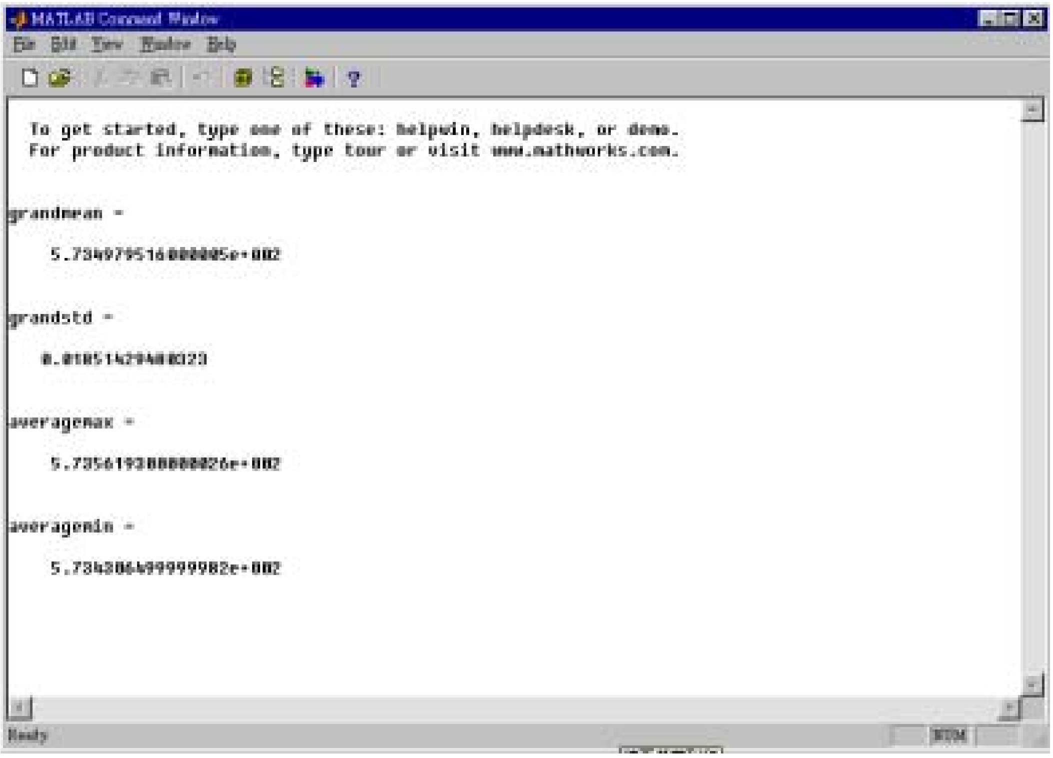
Figure 4 The MATLAB Program of Example 2

The notations of “average”, “samplestd”, “maximum”, and “minimum” are to compute sample average, sample standard deviation, the maximum value, and the minimum value, respectively, for each 100 resample data. For the grand mean, grand standard deviation, average maximum value, and average minimum value for these 1000 iterations, it is quite simple to use the built-in functions of mean, std, max, and min with appropriate data sources. For example, to calculate the grand mean, the built-in function “mean” is used with the data source of “average”, which is a row vector with 1000 values.

Finally, the results were generated by less than 10 seconds and are concluded in Figure 5. The grand mean, grand standard deviation, average maximum value, and average minimum value are 573.4980, 0.018514, 573.56193, and 573.43065, respectively. Obviously, using MATLAB programs makes the bootstrap method much easier and simpler in implementation and computation.

4. Conclusions

This paper shows that using MATLAB programs makes the bootstrap method much easier in implementation. In the first example, the MATLAB approach is much simpler in computing the same results. The second example uses a data set for the bootstrap method to generate some basic statistical calculations without any theoretical assumptions. Clearly, this technique has provided those who are interested in the bootstrap method without strong backgrounds in programming an effective way to learn and implement it with some easy MATLAB codes. On the contrary, those who are familiar with MATLAB can further study the properties of the raw data by re-writing M-files or using more sophisticated built-in functions in data analysis.



```
 To get started, type one of these: helpwin, helpdesk, or demo.
 For product information, type tour or visit www.mathworks.com.

grandmean =

    5.734979516000005e+002

grandstd =

   0.01851429480323

averagemax =

    5.735619380000026e+002

averagemin =

    5.734306499999982e+002
```

Figure 5 The Results of Example 2

References

1. Borse, G.J. (1997), *Numerical Methods with MATLAB: A Resource for Scientists and Engineers*, PWS Publishing Company, Boston, MA.
2. Efron, B. (1979), “Bootstrap Methods: Aother Look at the Jackknife,” *Annals of Statistics*, 7, pp. 1-6.
3. Efron, B. & R.J. Tibshirani, (1993), *An Introduction to the Bootstrap*, Chapman and Hall, New York.
4. Franklin, L.A. & G. Wasserman, (1991), “Standard Bootstrap Confidence Interval Estimates of C_{pk},” *Computers and Industrial Engineering*, 21, pp. 129-133.
5. Schilling, R.J. & S.L. Harris, (1999), *Applied Numerical Methods for Engineers Using*

MATLAB and C, Brooks/Cole Publishing Company.

6. Woodroof, J. (2000), “Bootstrapping: as easy as 1-2-3,” *Journal of Applied Statistics*, 27(4), pp. 509-517.
7. Wu, Z. & Q. Wang, (1996-97), “Bootstrap Control Charts,” *Quality Engineering*, 9(1), pp. 143-150.
8. Wu, H.-H. (2000), “Using the Johnson System in Clements-based Process Capability Indices for Non-Normal Processes,” in *Proceedings of the 5th Annual International Conference on Industrial Engineering – Theory, Applications and Practice*., No. 179, December 13-15, Hsinchu, Taiwan.
9. Wu, H.-H. (2001), “A Matlab Approach to Evaluate Product Quality,” *The Asian Journal on Quality*, 2(2), pp. 34-45.