USE OF THE SAS MACRO LANGUAGE IN DEVELOPING CONTROL CHART LIMITS
T. P. Walker, IBM General Technology Division, East Fishkill

ABSTRACT

Control charts are an important tool in applied statistics, especially in quality control. This report describes the generation of control chart limits for process data in which the limits are developed by first grouping the raw process readings into many relatively small groups and then, from this grouped data, obtaining subsets, which are stable with respect to the standard deviation and the mean.

An important feature of this report is the demonstration of the effective use of the SAS Macro Language in facilitating the calculations required to develop the stable subsets and the censor limits. Without the macro language capabilities, the computations would be long, tedious, and error-prone.

An example is given, and the SAS Macros, which make up the program, are listed.

I. INTRODUCTION

Control charts are a well-known tool in quality control, and the development of control chart limits is also fairly well known. However, both the concept of a statistically stable set of data and the procedure for generating such a set are less well known.

This paper demonstrates the use of the SAS Macro Language (Ref. 1, Chapter 15) in achieving the following objectives:

A) The generation - from a set of raw data - of a subset, which is statistically stable with respect to the standard deviation.

B) The generation of another subset, which is statistically stable with respect to the mean.

C) The development of control chart limits for individual values.

D) The generation of a subset of the original raw data readings which is in statistical control.

Ordinarily, a series of repetitive calculations would be required to achieve objectives A to D. Programs would have to be rerun several times. For each run, a user would make a decision based on the results of the previous run. However, the SAS Macro Language now makes it possible to carry out the computations in one job involving a series of repetitive calculations but without intervention by the user.

In section II, the steps leading to the development of the control chart limits are detailed. Sections III and IV contain an example as well as an annotated listing of the SAS Macros, which make up the program.

The statistical theory underlying the generation of control chart limits both for the process mean and for the process variation is presented in Ref. 2, Chapters 3 and 4. See also Ref. 3, Chapters 5, 6, and 7, for an even more detailed presentation of control chart theory.

The calculations presented in the next section and the example in the last section make use of a sample of size 9. As will be seen, the use of a specific sample size makes it possible to identify exactly the scaling factors, which are required for generating the stable sets of data and the control chart limits. It is hoped that the use of such specific values will help to clarify the presentation.

II. CONTROL CHART TECHNIQUES

This section describes the generation - from a set of raw process readings - of a subset of data, which is in statistical control. There is a series of 9 steps involved in the procedure. As will be seen, some of the steps may be repeated. The sequence proceeds from generating a set of data, which is stable with respect to the standard deviation, to generating a set stable with respect to the mean, to calculation of control chart limits for individual values, to generation of the subset, which is in statistical control.

1) Assume that N raw process readings are available. Let variable CT denote an individual reading. Divide the N CT-values into A groups, 9 members per group, i.e., A = N/9 rounded to the nearest integer.

2) Compute statistics CTS and CTS for each of the A groups, where

CMT = the mean of the set of CMTs
CMT = the standard deviation of the set of CMTs

3) Compute MCTS and MCTS, where

MCTS = the mean of the set of the A CMTs
MCTS = the mean of the set of the A CTSs

Put ULSTD = 1.761 X MCTS
and LLSTD = 0.239 X MCTS

858
The factors 0.239 and 1.761 are associated with the sample size of 9 and are obtained from Table B, in the Appendix of Ref (2).

4) From the A groups, delete any group whose standard deviation value \( \sigma \) is greater than \( \sigma_{ULSTD} \) or less than \( \sigma_{ULSTD} \). As a result, there will be \( A_1 \) groups left where \( A_1 < A \). If \( A_1 = A \), then objective A has been accomplished, i.e., we have a set of stable groups none of whose standard deviations, \( \sigma \), falls outside the control interval \( (\sigma_{LLSTD}, \sigma_{ULSTD}) \).

5) On the other hand, suppose that \( A_1 < A \). Then, using the \( A_1 \) groups, repeat steps 3 and 4. Compute new values for \( M_{CTM} \), \( M_{CTS} \), \( LLMTD \), and \( ULMTD \). Next, delete from the set of \( A_1 \) groups those whose \( \sigma \) falls outside the new interval \( (\sigma_{LLSTD}, \sigma_{ULSTD}) \). As a result, there will be \( A_2 \) groups left, where \( A_2 < A_1 \). If \( A_2 = A_1 \), then we are finished. Otherwise, repeat steps 3 and 4 using the \( A_2 \) groups. At some point, there will be no more deletions, i.e., \( A(K+1) = A_2 \) for some \( K \).

(Note: If all groups are deleted, then the process is totally out of control. We exclude this possibility.) At this point, a set of \( A_k \) groups stable with respect to the standard deviation has been obtained. Next, we develop a set stable with respect to the mean.

6) Let \( M_{CTM} \) = the mean of the set of the \( A_k \) \( CTM \)'s
\[ M_{CTS} = \text{the mean of the set of the } A_k \text{ } CTS \text{'s} \]
\[ \text{Put } ULMean = M_{CTM} + 1.094 \times M_{CTS} \]
\[ \text{LLMean} = M_{CTM} - 1.094 \times M_{CTS} \]
The factor 1.094 is obtained from Table B in the Appendix of Ref (2), based on the sample size of 9.

7) From the \( A_k \) group, delete any groups whose mean value \( \bar{X} \) falls outside the interval \( (LLMean, ULMean) \). There will be \( B_1 \) groups left where \( B_1 < A_k \). If \( B_1 = A_k \), then objective B has been accomplished. Otherwise, if \( B_1 < A_k \), then using the \( B_1 \) groups, repeat steps 6 and 7.

As before, at some point there will be no more deletions, i.e., \( B(T + 1) = B_1 \) for some \( T \). The \( B_1 \) groups are stable with respect to the mean. By construction, they are a subset of the \( A_k \) groups, which were stable with respect to the standard deviation. Hence, the \( B_1 \) groups are also stable with respect to the standard deviation. Thus, objectives A and B have been accomplished.

Now construct control chart limits for individual values and generate a subset of data which is in statistical control.

8) Let \( M_{CTM} \) = the mean of the set of the \( B_1 \) \( CTM \)'s.
\[ M_{CTS} = \text{the mean of the set of the } B_1 \text{ } CTS \text{'s} \]
\[ \text{Put } ULMean = M_{CTM} + (3/0.9139) \times M_{CTS} \]
\[ LLMean = M_{CTM} - (3/0.9139) \times M_{CTS} \]

As with the other factors, 0.9139 is associated with the sample size of 9 and is obtained from Table B in the Appendix of Ref. (2)

The control chart limits, \( ULMean \) and \( LLMean \) implement objective C.

9) From the set of \( N \) raw-process readings, delete any value \( \bar{X} \), which falls outside the interval \( (LLMean, ULMean) \). The \( N_1 \) values, which remain, represent a process, which is in control. These \( N_1 \) values implement objective B.

The next section presents an example of the 9-step procedure just described. The last section lists the SAS code, which implements the 9-step procedure.

III. EXAMPLE (USING THE NOTATION FROM SECTION I)

\[
\begin{align*}
3452 & = N = \text{Number of process readings} \\
384 & = (3452/9) = A = \text{Initial number of groups} \\
230 & = A_k = \text{Set of groups stable with respect to the standard deviation} \\
13 & = K = \text{Number of iterations required} \\
204 & = \text{Set of groups stable with respect to the mean} \\
5 & = T = \text{Number of iterations required} \\
8656 & = M_{CTM} = \text{Value for the initial set of 384 groups} \\
974 & = M_{CTS} = \text{Value for the 384 groups} \\
9265 & = M_{CTM} = \text{Value for stabilized set of 204 groups} \\
604 & = M_{CTS} = \text{Value for stabilized set of 204 groups} \\
11248 & = ULMean = \text{Upper control chart limit for individual values} \\
7282 & = LLMean = \text{Lower control chart limit for individual values} \\
2915 & = N_1 = \text{Subset of the original set} \\
2915 & = \bar{X} = \text{Mean value of CT for the original set of 3452 readings} \\
8656 & = \bar{X} = \text{Mean value of CT for the original set of 3452 readings} \\
9167 & = \text{Standard deviation of CT for the original set} \\
672 & = \text{Standard deviation of CT for this subset of 2915}
\end{align*}
\]
The program consists of a main calling routine and 4 SAS Macros: CENSOR, STABLEMN, REPEATER, and INPUTR. The code is listed at the end of this section.

Briefly, the programming sequence as follows:

1. Main routine calls INPUTR;
2. INPUTR calls REPEATER;
3. REPEATER calls STABLEMN;
4. Main routine calls CENSOR.

The main calling routine (lines 20500 to 21400) gets the input data, performs a few operations, and calls INPUTR.

The sample size of 9 and its associated scaling factors are arguments for INPUTR. These arguments are passed along to the other macros, as required. Macro INPUTR groups the data into sets of 9 (step 1 of II) and calls REPEATER. Macro REPEATER implements steps 2 to 5 of II. Notice that REPEATER will rerun until a stable set of data for the standard deviation is obtained or until the number of tries exceeds a user-defined bound (in this case, 20). When REPEATER finishes, it calls STABLEMN, which implements steps 6 and 7 of II. Like REPEATER, STABLEMN repeats until a stable set of data for the mean is obtained, or until the number of tries is exceeded.

When STABLEMN is finished, control returns to the main program, which calls CENSOR. Macro CENSOR implements steps 8 and 9 of II. Additionally, a few other statistics and graphs are generated. Notice that the arguments for CENSOR are related to the sample size of 9.

A listing of the program code follows.
IF H > 15 THEN DO;
   TOL preferably 0.3;
   CALL SYMBOL('TOL', 0.3);
END;
ELSE DO;
   TOL preferably 15;
   CALL SYMBOL('TOL', 15);
END.

REFERENCES
2. R. C. Vaughn, Quality Control, Iowa State University Press, Ames, Iowa, 1974

TRADEMARK NOTICE
SAS is a registered trademark of the SAS Institute Inc., Cary, NC, USA.

For More information, contact:
Thomas P. Walker Jr.
IBM Corporation, East Fishkill Facility
Dept. 165, Zip 63A, Route 52
Hopewell Junction, New York 12533
914-894-5293