SESUG Paper 51-2021

A Macro to Utilize a Nonparametric Multiple Stream Process Quality Control Chart in SAS

Austin R. Brown, Ph.D., Kennesaw State University

ABSTRACT

Statistical process control charts have been shown to be useful tools in monitoring and improving the quality of a variety of processes over the past century. For a control charting scheme to be successful, it should be chosen to match the specifications of the process to be monitored. For example, there may be instances when several processes which are assumed to be identical and desired to be monitored simultaneously rather than independently. Such a process is commonly referred to as a "multiple stream process (MSP)." Control charts designed for MSP monitoring have typically assumed that the observations being monitored follow a Normal distribution. This assumption may not always be met, in which case, the existing charts begin to become inefficient at detecting anomalies in the process being monitored. Recently, a new control chart was developed for monitoring MSPs, which is nonparametric in nature, which means that its performance will remain consistent regardless of the underlying distribution. This chart is called the "Nonparametric Extended Median Test Cumulative Summation Chart (NEMT-CUSUM)." However, one issue with this control chart is that there is no current procedure for utilizing the technique in SAS software. Thus, the purpose of this paper is to develop a SAS macro function to use the NEMT-CUSUM control chart in SAS. Examples and discussion will be provided.

INTRODUCTION

Statistical process control is a set of procedures used to systematically monitor some sort of process or processes. The main goal of these procedures is to improve quality by quickly identifying and ameliorating causes of poor quality. A specific tool developed to help in this goal is the control chart. Originally developed by Walter Shewhart in the 1920s, control charts provide the user with a graphical tool to compare some summary statistic (e.g., sample mean, sample variance, etc.) or function of a summary statistic (commonly referred to as a "plotting statistic") obtained from the process itself to control limits derived from the sampling distribution of the summary statistic (Shewhart, 1924). If the summary statistic exceeds a control limit, then the process is said to be "out of control" and an investigation into the cause for this out-of-control (OOC) point begins. Obviously, if the process is OOC, the chart operator will want to know that as quickly as possible, which operationally means in as few samples as possible. Over time, the variety of processes practitioners desired to monitor grew in diversity and complexity, which required the development of new charting schemes. Woodall and Montgomery (2014) give a good overview of many recent advances. The particular control chart one uses should be one best suited for the specific attributes of the process to be monitored. For example, one may be interested in monitoring several presumably identical and independent processes, like call center wait times per customer service representative or number of transactions per bank teller. In process control literature, such a process is typically referred to as a "multiple stream process (MSP)" (Montgomery, 2013).

There have been several control charting schemes developed specifically for the monitoring of MSPs (e.g., Boyd, 1950; Mortell & Runger, 1995; Meneces et al, 2008; Vicentin et al, 2018). However, all of these control charts rely on the assumption that the data being collected from the process follows a Normal or Gaussian distribution. As is the case with statistical methods also relying on the assumption of normality, when this assumption is not reasonably met, the ability of the control chart to quickly detect OOC points is deteriorated (Chakraborti & Graham, 2019a). When this is the case, it is typically recommended to use a "nonparametric" control chart, which is one that does not have any distributional assumption. Since non-normality is a reality for lots of processes, there have been many proposed nonparametric control charting schemes in recent years. Chakraborti and Graham (2019b) give a thorough overview of many advances in the arena of nonparametric control charting, specifically. Considering that practitioners may be in need of a nonparametric MSP control chart, Brown and Schaffer (2020) proposed one which utilizes a function of a classical nonparametric test statistic as its plotting statistic. This chart is referred to as the "Nonparametric Extended Median Test Cumulative Summation

Chart (NEMT-CUSUM).” Through simulation, this chart was shown to have more consistent and, in several instances, improved performance regardless of distribution compared to its parametric counterparts (Brown & Schaffer, 2020). However, to date, there has been no formal way of implementing this chart in any statistical software package, SAS or otherwise. Thus, the purpose of this paper is to demonstrate how and why this chart works through the explanation of the components of a SAS macro developed for this chart. This macro is called “`%countr`.”

FICTICIOUS MONITORING SITUATION

For the purpose of this paper, let’s assume that we are interested in monitoring wait times (in minutes) for 10 customer service representatives at a call center. Suppose during a given work week, we randomly sample 10 customer wait times per representative. In total, for a given week, we would have n = 400 observations per representative. Let’s also assume that we target a median wait time of 2 minutes.

DATA STRUCTURE

One important consideration for any statistical analysis is the determination of how the data need to be organized or structured in order for a function, PROC, or macro to work correctly. The way `countr` is designed, each stream (customer service representatives, in this case) should be an individual column. An additional column is required to indicate the sample number or sampling interval (in our case, hour of the work week) and must be appended to the left of the data table. Further, each column needs to be sequentially named (e.g., rep1, rep2, rep3, etc). For us, the data used in this example are structured as shown in Figure 1.

Obs	hour	rep1	rep2	rep3	rep4	rep5	rep6	rep7	rep8	rep9	rep10
1	1	1.55788	0.01848	1.53522	0.44440	0.5570	3.67303	1.25656	2.78176	2.3144	0.3833
2	1	1.89885	0.49470	3.69562	0.19874	4.8395	2.88310	6.62371	0.86554	5.4696	0.5017
3	1	0.82975	0.01820	6.84760	0.87642	1.3708	1.03070	1.54070	2.56100	0.1714	12.3152
4	1	4.68049	0.83282	0.71419	1.62080	2.1384	0.07965	0.03997	0.49569	0.5646	3.9063
5	1	0.59971	1.89383	2.45170	2.68946	0.9136	0.60352	0.97118	3.52688	0.5777	3.0023
6	1	4.86587	2.83655	1.64880	2.92272	10.5738	1.72041	0.70794	7.99259	12.2116	0.8283
7	1	4.11816	1.04065	0.40213	0.12877	1.0945	5.89013	0.02099	4.25849	2.0969	0.6272
8	1	0.70602	7.94051	1.29149	0.87377	0.1015	2.92246	1.12553	3.32337	1.3737	5.5973
9	1	1.50541	0.20542	3.24176	4.70358	0.6957	0.10237	2.35495	0.03178	2.3403	3.2421
10	1	1.92140	5.11076	1.30124	0.87276	0.4710	0.18566	6.97717	2.23363	1.9341	4.7667

Figure 1: Example of Data Structure Necessary for Macro to Function Properly

NEMT-CUSUM CHART CONSTRUCTION

COUNTING OBSERVATIONS GREATER THAN THE MEDIAN

Nonparametric methods commonly rely on the monitoring of a location parameter, typically the median, instead of the mean. This is done because if some specific value is really the median of some process, then we would expect, on average, that about 50% of the observations would be greater than this value and about 50% of the observations would be less than this value. With the NEMT-CUSUM chart, we assume that we have some known quantity for the process median (denoted $\tilde{\mu}$ and in our example in 2 minutes). At each time point, we take a sample of at least n = 10 for each stream being monitored (suppose in general there are C-many streams) and count up how many meet or exceed the known median (assuming our data are continuous). At each time point, t, we can conceptualize and generalize this counting process as given in Table 1.

Stream Number	1	2	...	C
Number of Observations $\geq \tilde{\mu}$	O_{1t}	O_{2t}	...	O_{ct}
Sample Size	n_{1t}	n_{2t}	...	n_{ct}

Table 1: Counting Observations Meeting or Exceeding the Median

Since we are assuming the streams are mutually independent, we can consider each O_{it} a binomial random variable with mean and variance of:

$$E[O_{it}] = n_{it}(0.50) \qquad (1)$$
$$Var[O_{it}] = n_{it}(0.50)^2.$$

Note, while theoretically the sample sizes for each stream do not have to be equal, for the `countr` macro, they do need to be equal. Now the question becomes, "How do we implement this counting process in SAS?" This was done with the help of PROC SQL. The code below gives an example of how this process works:

```
proc sql;
    create table ex as
    select hour, rep1 from waiting;
    create table ex1 as
    select * from ex
    where rep1 > &mu0;
    create table ex2 as
    select hour,count(*) as frequency
    from ex1
    group by hour;
quit;
```

Note, the macro variable `&mu0 = 2`. The first table created by PROC SQL separates each individual stream and the sample number (see Figure 2).

Obs	hour	rep1
1	1	1.55788
2	1	1.89885
3	1	0.82975
4	1	4.68049
5	1	0.59971
6	1	4.86587
7	1	4.11816
8	1	0.70602
9	1	1.50541
10	1	1.92140

Figure 2: Table "ex" from PROC SQL Code

The second table created by PROC SQL filters the first table such that only observations which are greater than the target median, `&mu0`, are retained. The third and final table created by PROC SQL counts up those unique rows for a given stream and a given sample number which are greater than the target median. This final table is given in Figure 3.

Obs	hour	frequency
1	1	3
2	2	4
3	3	6
4	4	6
5	5	4
6	6	5
7	7	8
8	8	5
9	9	1
10	10	5

Figure 3: Table "ex2" from PROC SQL Code

Now, this process had to be generalized and automated for up to C-many streams. To do this, there first had to be a way to get the column names from the data table into a macro variable. A macro named `get_cols` was implemented to do this very thing:

```
%macro get_cols(lib,mem,mvar,type);
   %global &mvar;
   proc sql noprint;
      select
         name
      into
         :&mvar separated by ' '
      from
         dictionary.columns
      where
            libname eq upcase("&lib")
         and memname eq upcase("&mem")

         %if %upcase(&type) ne ALL %then
         and upcase(type) eq upcase("&type");
      ;
   quit;
   %put &mvar = &&&mvar;
%mend get_cols;
```

In this macro, “lib” is the library where the data table in stored, “mem” is the name of the data table itself, “mvar” is what you decide to name the macro variable containing the column names, and “type” refers to the type of variable you want the name of (e.g., numeric, character, or all). For the given data table, we can see that this macro executes properly by executing the following code:

```
%get_cols(WORK,waiting,vars,all);
```

and obtaining the following SAS log message:

```
vars = hour rep1 rep2 rep3 rep4 rep5 rep6 rep7 rep8 rep9 rep10
```

To loop through this process, the following code was used:

```
%do i=2 %to %sysfunc(countw(&vars));
   %let col_num = %scan(&vars, &i);
   %let name2 = %sysfunc(catx(_,&col_num,1));
   %let name3 = %sysfunc(catx(_,&col_num,2));
   proc sql;
   create table &col_num as
   select &time_var, &col_num from &df;
   create table &name2 as
   select * from &col_num
   where &col_num > &mu0;
   create table &name3 as
   select &time_var,count(*) as frequency
   from &name2
   group by &time_var;
   quit;
   data &name3;
   set &name3;
   ID = "&col_num";
   run;
%end;
```

Walking through this code, we first must automate a way for SAS to know how many streams our process has. To do this, the `countw` function was used. It counts the number of words contained in a macro variable. In this case, that number is 11, one for each of our streams plus the sample number variable. Because we also have the sample number variable, the DO loop must start iterating at i=2 (rep1) rather than i=1 (hour). The first LET statement sets the macro variable `col_num` equal to the name of a particular stream. So when i=2, `col_num` = rep1. The next two LET statements concatenate "_1" and "_2" respectively to the end of the `col_num` macro variable. The DATA step at the end of the DO loop appends a stream ID variable to the data table created as in Figure 3. These "_2" data tables are then vertically joined by first identifying them in the WORK library and then by using the SET statement in the DATA step:

```
proc sql noprint;
      select distinct memname
      into :dfnames separated by " "
      from dictionary.columns
      where libname eq "WORK" and memname contains "_2";
quit;
```

```
data nemt_cusum_file;
set &dfnames;
run;
```

CALCULATING THE PLOTTING STATISTIC AND CONTROL LIMITS

After determining the number of observations greater than the median for each stream at each time point, Brown and Schaffer (2020) suggest standardizing the frequencies. For a sufficiently large sample size ($n_{it} \geq 10$), each of these standardized frequencies will approximately follow a standard normal distribution:

$$Z_{it} = \frac{O_{it} - n_{it}(0.50)}{\sqrt{n_{it}(0.50)^2}} \dot{\sim} N(0,1). \tag{2}$$

Summing these frequencies yields a quantity defined as:

$$EMT_t = \sum_{i=1}^{t} Z_{it} \dot{\sim} N(0, C). \tag{3}$$

Thus, at each time point, *t*, we will obtain an EMT statistic. If the process is in-control, which is to say that the value of $\tilde{\mu}$ really is the process median, we would expect the EMT statistics to be around 0. If the process is OOC, the values of EMT will begin to substantially deviate from 0. The plotting statistic for the NEMT-CUSUM chart is the cumulative sum of all previously observed EMT statistics:

$$S_t = \sum_{j=1}^{t} EMT_j = S_{t-1} + EMT_t, \tag{4}$$

where $S_0 = 0$. Since it is clear that S_t depends on S_{t-1}, Brown and Schaffer (2020) justified the use of the conditional distribution of $S_t|S_{t-1}$ which they showed to be approximately:

$$S_t|S_{t-1} \dot{\sim} N(s_{t-1}, C). \tag{5}$$

The recommended control limits are:

$$\begin{aligned} UCL_t &= s_{t-1} + \delta\sqrt{C} \\ LCL_t &= s_{t-1} - \delta\sqrt{C}, \end{aligned} \tag{6}$$

where δ is the "half-width" (typically taken to be 3 such that the probability of a "false alarm," observing an OOC point when the process is really in-control, is the traditional value of 0.00027). Now here, if the process is in-control S_t should be around 0. If the process is OOC, S_t will drift away from 0 in either the positive or negative direction. The control limits provide us with a "cut point" to determine when a drift away from 0 is substantial enough to warrant investigation and perhaps corrective action. To obtain all of this information in `countr`, the following code was implemented:

```
data nemt_cusum_file;
set nemt_cusum_file;
Zt = (frequency - %sysevalf(&n*0.50))/(%sysevalf(&n*0.25)**0.50);
run;
```

```
proc sql noprint;
create table np_plot as
select &time_var, sum(Zt) as EMT
from nemt_cusum_file
group by &time_var;
quit;

data np_plot;
set np_plot;
retain St;
St+EMT;
St1 = lag(St);
if St1 = . then UL = %sysevalf(&delta*(&&C)**0.50);
else UL = St1 + %sysevalf(&delta*(&&C)**0.50);
if St1 = . then LL =  %sysevalf(-1*&delta*(&&C)**0.50);
else LL = 2*St1 - UL;
/* Any OOC points to identify? */
if St > UL or St < LL then OOC = hour;
run;
```

In the first DATA step, the Z_{it} statistics are calculated for each stream and each time point. In the PROC SQL code, the EMT_t statistics are calculated for each time point. Then in the final DATA step, the S_t statistics and the control limits are calculated. Additionally, another column named "OOC" is created which shows if a particular time point plotted OOC. This last data table, np_plot, is similarly structured to data tables outputted by procedures such as PROC SHEWHART, PROC EWMA, and PROC CUSUM. Figure 4 gives an example of what this data table looks like:

Obs	hour	EMT	St	St1	UL	LL	OOC
1	1	-5.69210	-5.6921	.	9.48683	-9.4868	.
2	2	-3.16228	-8.8544	-5.6921	3.79473	-15.1789	.
3	3	1.26491	-7.5895	-8.8544	0.63246	-18.3412	.
4	4	5.05964	-2.5298	-7.5895	1.89737	-17.0763	.
5	5	-2.52982	-5.0596	-2.5298	6.95701	-12.0167	.
6	6	-3.16228	-8.2219	-5.0596	4.42719	-14.5465	.
7	7	-1.89737	-10.1193	-8.2219	1.26491	-17.7088	.
8	8	2.52982	-7.5895	-10.1193	-0.63246	-19.6061	.
9	9	-1.89737	-9.4868	-7.5895	1.89737	-17.0763	.
10	10	-1.89737	-11.3842	-9.4868	0.00000	-18.9737	.

Figure 4: Final NEMT-CUSUM Macro Data Table

This table shows the end user every piece of information calculated at each step as previously described. The values in the OOC column are coded as missing unless an observation plots OOC. Finally, because

this information is typically plotted in a graph, the countr macro also provides the user with a nice graph generated by PROC SGPLOT and is given by Figure 5:

```
proc sgplot data=np_plot;
series x=&time_var y=St/markers;
series x=&time_var y=UL/markers;
series x=&time_var y=LL/markers;
xaxis integer ;
run;
```

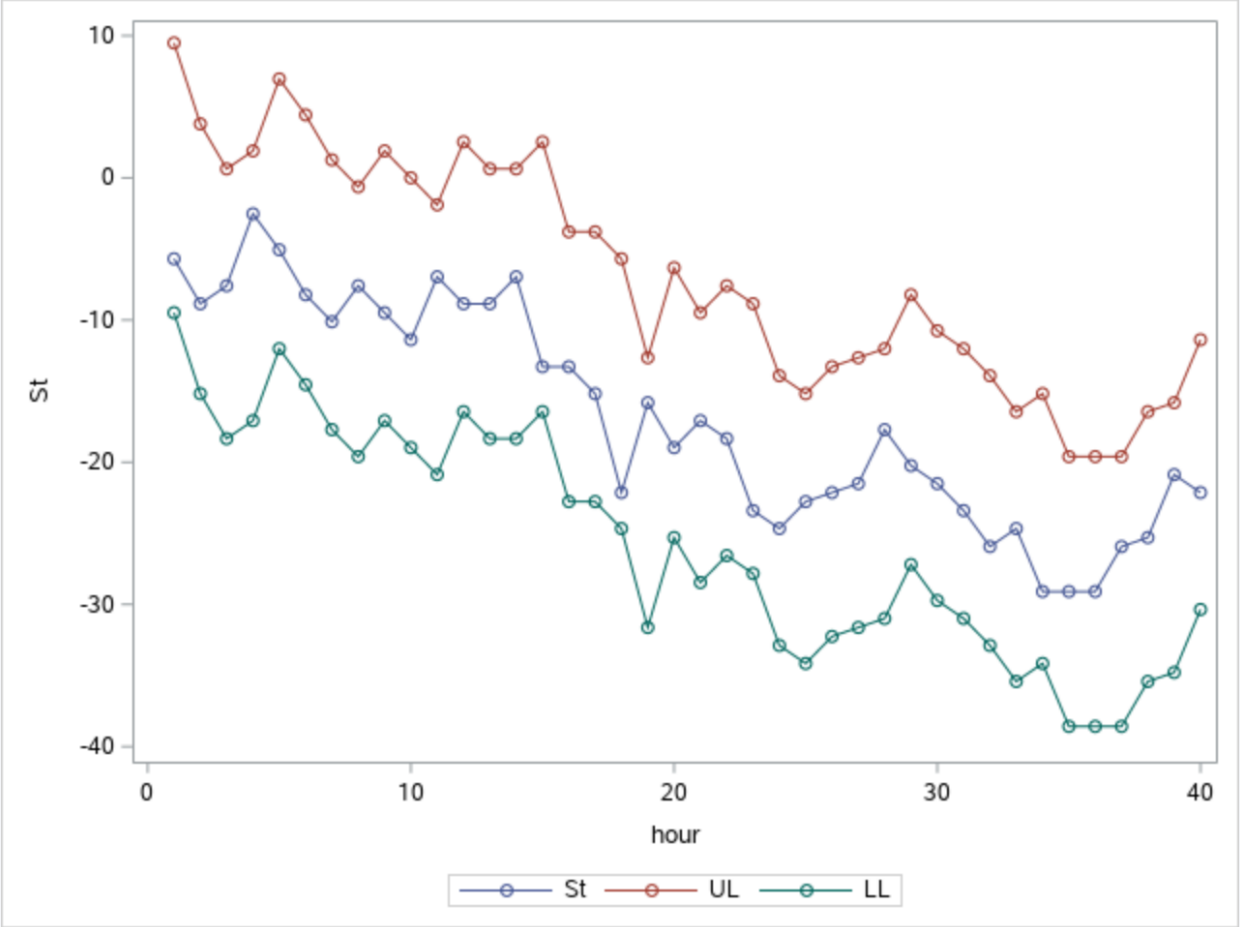

Figure 5: NEMT-CUSUM Chart Example

Since all of the values of S_t plot within the control limits, this would indicate that the process is in-control. In other words, we do not have significant evidence to suggest the median wait time differs from 2 minutes.

PUTTING IT ALL TOGETHER: USING THE COUNTR MACRO

The countr macro, after being read into SAS memory, can be executed as:

```
%countr(df,n,c,delta,mu0,time_var,lib,mvar,type);
```

In this macro, the position arguments are:

- df – the name of your data table, structured as Figure 1

- n – the size of the samples taken at each stream and each time point
- c – the number of monitored streams
- delta – the "half-width" of the control limits, typically taken to be 3
- mu0 – the in-control value of the process median
- time_var – the name of the sample number variable in your data table ("hour" in the working example)
- lib – the library where your data table is saved
- mvar – the name you choose for the macro variable which contains the column names of the data table
- type – the "type" of variable (e.g., numeric, character or all) you want the column names for stored in the macro variable mvar – generally "all" is the preferred choice

For the whole prior example, the following code was used:

```
%countr(waiting,10,10,3,2,hour,WORK,vars,all);
```

ADDRESSING AN OUT-OF-CONTROL POINT

When the process is in-control, meaning that S_t plots inside of the control limits, the user has assumes that all of the streams are in-control, meaning that the target value of the process median has likely not changed. When there is an OOC point, this means that *at least* one stream may have shifted away from target. In this case, the specific value of S_t is less meaningful in determining which stream or streams may be OOC. Brown and Schaffer (2020) propose two solutions to this problem.

First, the authors recommend identifying if a clear inflection point exists, one which immediately precedes a clear upward or downward pattern ultimately leading to the OOC point. All of the Z_{it} and individual frequency values for each stream at each time point from the inflection point to the OOC point should be examined to see which stream(s) are observed to have substantially large or small Z_{it} values. This can help the chart operator with their investigation into the identification of assignable causes.

Second, if there are no obviously large or small Z_{it} values for the stream(s), the authors recommend summing all of the frequencies for each stream from the inflection point up to and including the OOC point. Since it is assumed that a future sample taken from a given stream is independent from past samples taken from a given stream (i.e., no correlation between samples within or between observed streams), the sum of those frequencies, assuming the process is in-control, will also follow a binomial distribution with mean and variance:

$$E\left[\sum_{k=t_{Shift}}^{t_{OOC}} O_{ik}\right] = (0.50)\sum_{k=t_{Shift}}^{t_{OOC}} n_{ik}$$
$$Var\left[\sum_{k=t_{Shift}}^{t_{OOC}} O_{ik}\right] = (0.50)^2\sum_{k=t_{Shift}}^{t_{OOC}} n_{ik}, \quad (7)$$

where O_{ik} is the *i*th stream, t_{Shift} is the time point where the inflection point was noticed, and t_{OOC} is the time point where the OOC point was observed. These summed frequencies can be standardized and squared such that they asymptotically follow a $\chi^2(1)$ distribution, and p-values can be obtained for each χ^2 statistic. This strategy has not been studied in depth to determine if it is effective at correctly identifying OOC streams and thus should be taken as just one possible method for solving this problem, not necessarily the only method.

CONCLUSION

Statistical process control charts are effective tools to use in process monitoring applications. Because of the variety of processes, one should be sure to choose a chart best suited for the specific process being monitored. If one is monitoring a process which could be considered a MSP, it is advisable to use a chart designed for monitoring MSPs. When the assumption of normality cannot be reasonably assumed for a MSP, Brown and Schaffer (2020) recommend using their NEMT-CUSUM control chart, which does not have a distributional assumption at all. In this paper, a macro was developed and described for implementation in SAS. This macro provides the end user with an outputted dataset similar to those output by PROC SHEWHART, PROC CUSUM, and PROC EWMA as well as a graphical depiction of the chart.

REFERENCES

Boyd, D.F. (1950). Applying the group control chart for $\bar{x}$ and R. Industrial Quality Control, 7(3), 22-25.

Brown, A. R., and J. R. Schaffer. 2020. A nonparametric CUSUM control chart for multiple stream processes based on a modified extended median test. Communications in Statistics - Theory and Methods. Advance online publication. doi:10.1080/03610926.2020.1738492.

Chakraborti, S., and M. A. Graham. 2019. Nonparametric statistical process control. Hoboken, NJ: Wiley.

Chakraborti, S., & Graham, M. A. (2019). Nonparametric (distribution-free) control charts: An updated overview and some results. *Quality Engineering*, *31*(4), 523-544.

Meneces, N.S., Olivera, S.A., Saccone, C.D., & Tessore, J. (2008). Statistical control of multiple-stream processes: a Shewhart control chart for each stream. Quality Engineering, 20(2), 185-194.

Montgomery, D. C. (2013). Statistical quality control: A modern introduction. Hoboken, NJ: Wiley.

Mortell, R., & Runger, G. (1995). Statistical process-control of multiple stream processes. Journal of Quality Technology, 27(1), 1-12. doi:10.1080/00224065.1995.11979554

Shewhart, W. A. (1924). Some applications of statistical methods to the analysis of physical and engineering data. The Bell System Technical Journal, 3(1), 43-87. doi:10.1002/j.1538-7305.1924.tb01347.x

Woodall, W., & Montgomery, D. (2014). Some current directions in the theory and application of statistical process monitoring. Journal of Quality Technology, 46(1), 78-94

Vicentin, D. S., Silva, B. B., Piccirillo, I., Bueno, F. C., & Oprime, P. C. (2018). Monitoring process control chart with finite mixture probability distribution. International Journal of Quality & Reliability Management, 35(2), 335-353. doi:10.1108/ijqrm-11-2016-0196

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Austin R. Brown, Ph.D.
Kennesaw State University
College of Computing and Software Engineering
School of Data Science and Analytics
abrow708@kennesaw.edu