Paper ST16

Using SAS® to Control Multistream Binomial Processes with a Chi-Squared Control Chart

Peter Wludyka, University of North Florida, Jacksonville, FL; Dan Cavey, Bank of America, Jacksonville, FL & Brett Friedlin, University of North Florida, Jacksonville, FL

ABSTRACT

A process control scheme for multistream binomial processes is described. A multistream process in one in which the streams are uncorrelated or very weakly correlated, as might arise for example when *J* units are in a single frame that proceeds along a production line. The chi-squared control chart, which can be used for either homogeneous (the streams have identical rates at which nonconforming units arise) or nonhomogeneous (the streams have different rates at which nonconforming units arise) processes, will be described and SAS® programs provided that can be used to estimate the quantiles of the control limit distribution, estimate the ARL distribution for process and stream shifts, and produce control charts for process control. That is, a complete tool kit will be provided for initiating and maintaining process control using the SAS® system.

INTRODUCTION

A multistream process is best introduced by means of an example. Consider an injection molding process that makes a plastic toy. Tubes running from a reservoir carry liquid plastic into molds in an apparatus (a frame) which contains six molds. Suppose for simplicity that one tube is used to fill each toy. Each mold location can be considered to be a "stream" and this will be referred to as a 6-stream process. After cooling and separation from the mold a toy can be inspected and classified as defective (nonconforming) or nondefective (conforming). A process quality control scheme is a methodology for determine whether the process is producing "conforming units" in a stable and predictable manner; that is, one may decide whether the process is in-control or out-of-control. The control schemes described here require that periodically samples of product be inspected. One might be concerned with two ways in which the process could cease to be in-control.

1. A process change that affects all of the streams uniformly
2. A change that impacts on one or more streams differentially

The first might correspond with to a poor grade of plastic being used; the second to a clog or other problem with one or more individual tube.

Note that identifying which type of out-of-control condition occurs will be useful for bringing the process back under control.

PROCESS MODELING AND CONTROL CHARTS

Control charts can be used to control processes (see Montgomery). In order to use control charts a statistical model for the process is required. Suppose that *n* frames are selected for inspection at the end of each epoch. Then each stream will be sampled *n* times also. If the streams are uncorrelated then the process is called a *J*-stream multistream process. A frequently useful process model then is that each stream is a binomial process with *n* trials and success probability p_j, where p_j is the proportion of items in stream *j* that are nonconforming. When p_j is the same for all the streams the process is a homogeneous multistream process. (See Jacobs and Wludyka (2002) for details). When one or more streams have different p_j the process is non-homogeneous. One general way to characterize the process is to define

$$p_j = p + s_j \ .$$

In this definition the process is homogenous if for all values of *j=1,…,J* the $s_j = 0$. The process is deemed in-control as long as the rates at which nonconforming units appear remain constant. The process is out-of-control if

1. The overall *p* changes
2. s_j (p_j) changes for one or more streams

For homogeneous processes one may employ runs rules to control the process (see Wludyka and Jacobs 2001, 2002). For homogeneous as well as non-homogeneous processes one may control the process by using:

1. a single overall *p*-chart
2. *J* individual *p*-charts (one for each stream).
3. a Chi-squared chart

A frequently used tool for detecting changes in a binomial process is the *p*-chart (see Montgomery). The sample proportion nonconforming is plotted on the chart at each epoch and a signal occurs whenever the sample proportion plots outside the *k*-sigma control limits. Typically *k = 3* since experience has shown this to be a useful control band width. In (2) it is unwise to set *k = 3* since the false alarm rate will usually be much higher than most quality engineers would find reasonable. That is, *k* should be greater than three. See Jacobs and Wludyka (2002) for methods for determining the proper *k*. Note that is it sufficient to plot only the largest and smallest of the *J* sample (stream) proportions on a single chart for (2) when one has a homogeneous process. This chart is called a group p-chart. The Chi-squared chart will be described in this paper.

Chi-Squared Control Chart

Suppose that the rates at which nonconforming units arise is know or has been estimated. At each epoch samples of size n are collected and the stream sample proportion nonconforming is estimated by

$$\hat{p}_j = {y_j}/{n} \quad \text{for } j = 1,...,J \qquad (1)$$

Define

$$z_j = \frac{\hat{p}_j - p_j}{\sqrt{\frac{p_j(1-p_j)}{n}}} \quad \text{for } j = 1,...,J \qquad (2)$$

$$W = \sum_{j=1}^{n} z_j^2 \qquad (3)$$

Note that for large *n*,

z_j is approximately $N(0,1)$

and

W is approximatetly χ^2_J (J degrees of freedom) (4)

At each epoch the statistic W can be plotted on a Chi-Squared Control Chart with the appropriate Upper Control Limit (UCL). That UCL should be chosen so that

$$P(W > UCL) \le \alpha \qquad (5)$$

where the target in-control Average Run Length (ARL) is

$$ARL_0 = 1/\alpha \qquad (6) \ .$$

The quantiles of *W*, which are used to determine the UCL, can be found using the Chi-Squared distribution (based on equation (5)) or Monte Carlo estimates of the empirical distribution of *W* that can be found using the SAS® program that follows.

Example

Consider a 4-stream process from which samples of size one hundred are to be taken (n = 100). The rates at which nonconforming units are produced are given below.

Stream	1	2	3	4
Rate	.11	.06	.15	.06

Suppose one wants an in-control ARL of 370. Then

$$\alpha = 1/ ARL_0 = 1/370 = .0027$$

Hence, using the chi-squared distribution, the UCL = 16.2512 since

$$P(W \le UCL) = P(\chi_4^2 \le 16.2512) = 1 - .9973 = .0027$$

Macro for the Empirical Distribution of W

The macro below calculates the Upper Control Limit (UCL) for a Chi-Squared chart. It takes a set of known p_i for *j=1,…,J,* a sample size, target in-control ARL, and Monte Carlo repetition count and produces the W Control Limit and a table of W values. Since the macro is fast users should do at least 100,000 Monte Carlo repetitions.

```
%macro wdist1(p1=,p2=,p3=,p4=,p5=,p6=,p7=,p8=,p9=,p10=,
              j=,n=,reps=,arl0=);
/*******************
         p's are the stream probabilities
         j = number of streams
         n = sample size per stream
         reps = number of monte carlo values of W
         arl0 = target in-control average run length
*********************/
data wstats;
         jstrs=&j;
         nsamples=&n;
         mctrials=&reps;

/* calculate the probability of the ARL */
quadrant = 1 - 1/&arl0;

array p{10} (&p1,&p2,&p3,&p4,&p5,&p6,&p7,&p8,&p9,&p10);
array sesq(10);

/* calulations for the W denominator */
do i=1 to jstrs;
sesq(i) = (p(i)*(1-p(i)))/nsamples;
end;

seed1=-111;
chi=cinv(quadrant,jstrs);

/* calculate the Ws and output them */
do rep =1 to mctrials;
         W=0;
         do jstep = 1 to jstrs;
                  call ranbin(seed1,nsamples,p(jstep),y);
                  pdiff = (y/nsamples)-p(jstep);
                  zsquare = (pdiff*pdiff)/sesq(jstep);
                  W = W + zsquare;
         end;
         output;
end;

/* proc print data=wstats; /* for debug */

/* Do proc sort, means and freq to get ordered W values */
/* NOTE:quadrant was included to pass on to the next data step
*/

proc sort data= wstats; by W;

title "W for p=&p1,&p2,&p3,&p4,&p5,&p6,&p7,&p8,&p9,&p10 n=&n
J=&j reps=&reps";
title2 "W Control Limit for ARL = &arl0 ";

proc means noprint; var W chi quadrant;

proc freq data=wstats noprint; tables W*quadrant /out=wdat;

run;

/* get the desired value */
data value;
         set wdat;

/* put the cutoff in percent precision */
         testquad = quadrant*100;

         /* cumulate the percentage */
         if _N_=1 then cumper=percent;
else cumper=cumper+percent;

         /* is the cumulative percentage at the cutoff? */
         /* NOTE: to just print a table comment out */
/* this if-do-end statement */
         if cumper > testquad then do;
                  keep W;  /* keep only W */
                  output;
                  stop;    /* stop the loop, we have our W */
         end;
         retain cumper;  /* keep the accumulator variable */

/* to print the W for the ARL in set simdata */
proc print data=value;
run;
/*  The following prints the entire W distribution*/
data valueall;
         set wdat;
         testquad = quadrant*100;
         if _N_=1 then cumper=percent;
                  else cumper=cumper+percent;
         arl = 1/(1-cumper/100);
         retain cumper;
         title "W Distribution Table";
proc print data=valueall ; var w cumper arl;
run;
/*  end of print W distribution */

/*  Removing this set of comments prints the Proc Freq Output
for W

title "Output from Proc Freq on W";
proc freq data=wstats; tables W;
run;

 end of Proc Freq Output Statements */
%mend wdist1;

/* Executing the macro below
   produces W control limit for ARL0 and Table */
%wdist1(p1=0.11,p2=0.06,p3=0.15,p4=0.06,p5=0.0,p6=0.0,p7=0.0,
p8=0.0,p9=0.0,p10=0.0,j=4,n=100,reps=100000,arl0=370);/**/
```

Example W Distribution Output

The first output from the macro is:

```
W for p=0.11,0.06,0.15,0.06,0.0,0.0,0.0,0.0,0.0,0.0 n=100 J=4 reps=100000        514
                  W Control Limit for ARL = 370     07:15 Thursday, June 13, 2002

                              Obs       W

                               1     17.4403
```

This identifies the Control Limit for the Chart. Then a Table of W

values (a partial tables appears below) is produced. Those who wish may comment this out, but this output is useful for those shopping for a control scheme (UCL). Observe that using the nominal Chi-squared control limit of 16.2512 produces an in-control ARL of about 246, which is well below the target of 370.

Obs	W	cumper	arl
2860	16.2226	99.593	245.700
2861	16.2518	99.594	246.305
2862	16.2525	99.596	247.525
2863	16.2672	99.597	248.139
2864	16.2684	99.598	248.756
2969	17.4339	99.728	367.647
2970	17.4403	99.730	370.370
2971	17.4562	99.731	371.747

PROCESS SHIFTS

Runs rules are of no value for detecting an overall process shift. For that reason it is often wise to use the runs rule in conjunction with another tool, such as a process *p*-chart, in which the stream data is aggregated to form a single *p*-chart. It is also possible to use a Chi-Squared chart with a runs rule or a group *p*-chart (see Wludyka and Jacobs (2002) for a thorough explanation). Using a control chart along with a runs rule will decrease all ARLs, especially the in-control ARL. so care must be used when employing such a control scheme.
The advantage of a dual scheme is that both stream shifts and overall shifts are more easily detected.

If one plans to use a *p*-chart with a runs rule it is probably wise to choose a Runs Rule with an in-control ARL of 500 or more. Then using a *p*-chart with $k = 3$ will probably likely produce a scheme with in-control ARL of about 225.

Simulation Tools for Chi-Square Charts

Monte Carlo methods can be used to estimate the *ARL*s associated with a complex control scheme. This macro can be used to simulate up to 10 non-homogeneous streams with unique shifts applied to each stream. Variables for controlling the sample size per epoch and the Monte Carlo repetitions are also included.

There is a sample code for using the macro follow the macro. It is used in combination with the macro presented earlier in the paper.

To use this macro:
Following the example at the end of the macro text below, you first call to the wdist1 macro defined earlier in this paper. This will create a data set called “value” that will contain the “W” that is the UCL for the Chi-Squared chart.

Now you can call the “arlcalc” macro as many times as you want to calculate the ARL for any situation of non-homogeneous streams up to 10. The macro will allow you to simulate shifts in a stream by entering a number in the appropriate corresponding shift parameter. The shift is entered as an integer, but the shift is implemented in thousandths. (i.e. a parameter of 50 would shift the stream *p* by 0.05.)

This complete example can be downloaded from the web site described later in this paper.

```
/*************************************************************
*/
/* do for these stream configurations
/*        do for these sample sizes
/*        calculate the overall P and
/*                 sigma numerator from the streams of interest
/*        calculate control limits for overall p-chart
/*        calculate individual stream Pj chart limits
/*        do for numreps i.e. do the monte carlo
/*            do until all signals have happened or 2000 reps
/*                generate random samples
/*                CALCULATE SIGNAL STATISTICS
/*            end
/*        end
/*        calculate ARLs
/*    end
/* end
/*************************************************************
*/
/* Parameters
/* the first 10 parameters are the "known" probabilties
/*        for each stream of interest
/* jstreams is the number of streams under consideration
/* samples is the number of samples at each "epoch"
/* repeat is the number of Monte Carlo repetition
/* the next 10 parameters are any stream shifts to consider
/*************************************************************
*/

%macro arlcalc(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,
                  jstreams,samples,repeat,
                  ss1,ss2,ss3,ss4,ss5,ss6,ss7,ss8,ss9,ss10);
data simdata;
set value;
clock = 111;
seed1 = 81*clock;
numreps = &repeat;

array sshift{10}
        (&ss1,&ss2,&ss3,&ss4,&ss5,&ss6,&ss7,&ss8,&ss9,&ss10);
array p0{10} (&p1,&p2,&p3,&p4,&p5,&p6,&p7,&p8,&p9,&p10);

/* assign any stream shifts that may have been requested. */
do i=1 to 10; sshift(i)=sshift(i)/1000; end;

array Zj(10);
array p(20);
array y(20);
array pjhat(20);
array sigp0j(20);

do jnum=&jstreams;   /* do for these stream configurations */
   do n=&samples;  /* do for these sample sizes */

      /* calculations for later */
      do i=1 to jnum;
         sigp0j(i)=sqrt((p0(i)*(1-p0(i)))/n);
      end;

      /* initialize the tsig  */
      tsig=0;

      /* shift the proportions as desired */
      do j=1 to jnum;
         p(j) = p0(j) + sshift(j);
      end; /* END DO j*/

      /* repeat for numreps i.e. do the monte carlo */
      do reps = 1 to numreps;

         /* Initialize signal */
         signal = 0;

         /* do until all signals have happened or 2000 reps */
         repct = 0;
         do until((signal ne 0) or (repct > 2000));
            repct = repct + 1;

            /* generate random samples */
            seed1 = seed1+1;
            do j=1 to jnum;
               y(j) = ranbin(seed1,n, p(j));
               pjhat(j)=y(j)/n;
            end; /*END DO j*/
```

```
            /* Calculate W* and do the test */
            pwtest=0;
            do j = 1 to jnum;
               Zj(j)=(pjhat(j)-p0(j))/sigp0j(j);
               pwtest=pwtest+(Zj(j)*Zj(j));
            end;
            if pwtest gt W and signal = 0
               then signal=repct;
         end; /* do until */

         /**** Here's were arls are summed for each rep */
         tsig=tsig+signal;

      end; /* end do reps*/

      /* calculate arls */
      arlw=tsig/numreps;

      output;

   end; /*end do n*/
end;/*end do jnum*/
run;

title "ALR Chi-squared (W)";
proc print;
   by jnum n;
   var numreps W arlw;
run;

%mend arlcalc;

/* Complete Example */

/* get the W for the ARL above */
%wdist1(p1=0.11,p2=0.06,p3=0.15,p4=0.06,p5=0.0,
       p6=0.0,p7=0.0,p8=0.0,p9=0.0,p10=0.0,
       j=4,n=100,reps=100000,arl0=370);/**/

/* call the arlcalc to get the in control info */
%arlcalc(0.11,0.06,0.15,0.06,0.0,0.0,0.0,0.0,0.0,0.0,
         4,100,10000,
         0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0);/**/

/* example of non-homogenous with shifts */
%arlcalc(0.11,0.06,0.15,0.06,0.0,0.0,0.0,0.0,0.0,0.0,
         4,100,10000,
         50.0,0.0,50.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0);/**/
quit;
```

Sample Output from the Simulator

Below you can see in the first print output gives the "in-control" ARL of 354.88. This is close to our expected ARL of 370. The second shows that with a shift of 0.05 in two of our non-homogenous streams the ARL is now 9.9746

```
            ALR Chi-squared (W)
-------------- jnum=4 n=100 -----------

 Obs     numreps        W         arlw
  1       10000     17.3844     354.88

            ALR Chi-squared (W)
-------------- jnum=4 n=100 -----------

 Obs     numreps        W         arlw
  1       10000     17.3844     9.9746
```

Conclusion

These macros can be used to calculate *ARL*s for many complex control schemes.

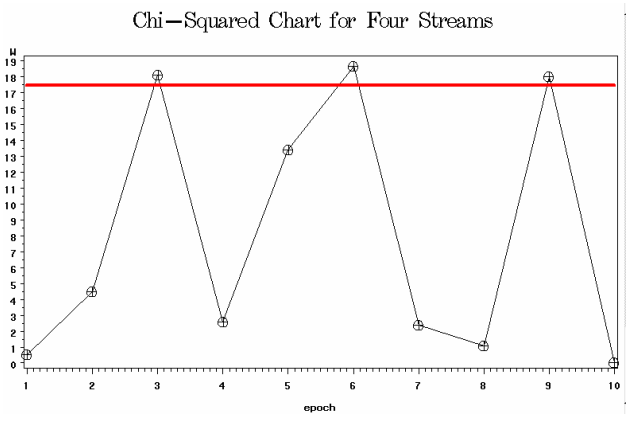
THE CHI-SQUARED CHART

The chart can be produced using the program that follows. It requires that the W statistic be calculated for each epoch. The program could be modified to do that. One could also modify PROC SHEWHART to produce attractive charts. The control chart limit is the one for the previous example as supposes that the in-control stream rates at which nonconforming units occur is that give previously.

SAS Program for Creating Chart

```
/*  This program uses a data stream in which the
       variable W has been calculated at each
epoch */
data streams;
input
epoch   stream1 stream2 stream3 stream4 W;
CL = 17.4403;
if W > CL then signal = 1; else signal = 0;
cards;
1       12      5       16      7       0.5352
2       13      6       11      10      4.5004
3       10      16      16      5       18.0884
4       15      7       14      8       2.5993
5       16      11      20      11      13.3797
6       18      11      21      12      18.6442
7       11      8       14      9       2.3834
8       12      7       16      4       1.0671
9       3       4       6       1       18.0321
10      11      6       15      6       0.0000
;
proc print;
run;
title f=centx 'Chi-Squared Chart for Four
Streams';
proc gplot;
       plot w*epoch=1
       cl*epoch=2 / overlay;
       symbol1 v=+ h=1.5 i=join c=black;
       symbol2 v=none w=4 i=join c=red;
       run;
```

Program Output



The interpretation is straightforward: out-of-control signals have occurred at epochs 3, 6, and 9 since the W statistic plots above the Control Limit (solid horizontal line at 17.4403). At the time of the signal a search for an assignable cause should commence.

DOWNLOADING SAS® PROGRAMS

Source code can be downloaded from the University of North Florida Center for Research and Consulting in Statistics web page (www.unf.edu/coas/math-stat/CRCS) as technical reports.

REFERENCES

Montgomery, D. C. (1997). *Introduction to Statistical Quality Control*, 3rd ed., John Wiley and Sons, New York.

Wludyka, P and Jacobs, S., "Runs Rules and P-Charts for Multistream Binomial Processes," *Communications in Statistics — Simulation and Computation*, 2002, pp97-142.

Wludyka, P and Jacobs, S., "Using SAS® to Control Multistream Binomial Processes," *Proceedings of the Joint Conference of the South East and South Central SAS® Users Groups*, New Orleans, LA, 2001, pp665-672.

CONTACT INFORMATION

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

Peter Wludyka
Associate Professor of Statistics
Director,
Center for Research and Consulting in Statistics
University of North Florida
Jacksonville, Florida
Work Phone: 904-620-1048
Fax: 904-620-2818
Email: pwludyka@unf.edu
Web: www.unf.edu/coas/math-stat/~pwludyka

Dan Cavey
Measures and Metrics Manager
Bank of America
Work Phone: 904-987-8128
Email: dan.cavey@bankofamerica.com

Brett Friedlin
Department of mathematics and Statistics
University of North Florida