On the computation of the principal constants d_2 and d_3 used to construct control limits for control charts applied in statistical process control

Joh. J. Sauren*

March 3, 2021

Abstract

In this communication a short and straightforward algorithm, written in Octave (version 6.1.0 (2020-11-26))/Matlab (version '9.9.0.1538559 (R2020b) Update 3'), is proposed for brute-force computation of the principal constants d_2 and d_3 used to calculate control limits for various types of variables control charts encountered in statistical process control (SPC).

1 Rationale of the computation

SPC is defined as a collection of tools that are essential in quality-improvement activities[1, 2]. To improve production or process quality, control–type charts are constructed and interpreted, in which control limits based on the so called d_2 and d_3 constants play a prime role. According to the best of the author's knowledge, mathematical formulae for the constants d_2 and d_3 in closed form are all but known or available. Usually, these constants are presented in tables as a function of the sample size n, i.e. as the number n of observations or measurements in a single sample. Frequently, a set of 20 to 25 samples is adopted to design a control chart based on the aforementioned constants. However, the tabulated constants are generally presented without reference as to what mathematical algorithm was utilized to arrive at their numerical figures. To alleviate dependency on tables, in the following an Octave/Matlab compatible algorithm is described that can be integrated in any type of SPC machine–based generation of different variables control charts.

2 The algorithm

A matrix M with $n_{rows} > 1 \cdot 10^7$ number of rows and $n_{columns} \geq 2$ number of columns is defined consisting of $n_{rows} \cdot n_{columns}$ random numbers. The randn command[3, 4] generates normally distributed numbers with mean 0 and standard deviation of 1. Prior to the generation of the matrix M, the rng shuffle command is used to reseed the random number generator with a different

*Department of Engineering, Zuyd University of Applied Sciences, NL–6419 DJ, Heerlen, The Netherlands, hans.sauren@zuyd.nl

seed[5]. As this command hasn't been implemented in Octave yet, it is recommended to omit it in the current code; running the code in Octave without the rng shuffle command does by no means impair the validity of the numerical results of the number punching. To compute the constant d_2, the mean of the ranges per each row for all the $n_{rows} > 1 \cdot 10^7$ rows is calculated, likewise, the constant d_3 is computed using the standard deviation of all ranges for each of the $n_{rows} > 1 \cdot 10^7$ rows.

```
clear
%
% omit the rng shuffle function in Octave as it has not been
% implemented yet
%
rng shuffle
%
% input
%
n_rows = 2e7;
n_columns = 2;
%
% preallocation of memory to speed up computation in for--loop
%
M = randn(n_rows, n_columns);
r = zeros(n_rows,1);
%
% computation of the range of each row
%
    for row = 1 : n_rows
        r(row) = max(M(row,:)) - min(M(row,:));
    end
%
% computation and output of the statistical process constants
%
n = n_columns
d2 = mean(r)
d3 = std(r)
A2 = 3/(d2*sqrt(n_columns))
D4 = 1 + 3*d3/d2
if n_columns > 6
    D3 = 2 - D4
else
    D3 = 0
end
```

3 Comparison of computational and literature results

The next table gives an impression of the accuracy of the algorithm when compared to tabulated literature values[1]. Calculations were run with n_{rows} =

$2 \cdot 10^7$ rows. The precision of the computated results varies in the last digit for each run.

n	computed d_2	literature[1] d_2	computed d_3	computed D_4	literature[1] D_4
2	1.1281	1.128	0.8527	3.2666	3.267
3	1.6926	1.693	0.8882	2.5739	2.575
4	2.0582	2.059	0.8795	2.2820	2.282
5	2.3252	2.326	0.8646	2.1155	2.115

References

[1] Douglas C. Montgomery, George C. Runger, Norma Faris Hubele, Chapter 8, Statistical Process Control, *Engineering Statistics, SI Version, 5ed*, Wiley, 2012, p. 438 – 482 and Table VII, Factors for Constructing variables Control Charts, p. 499.

[2] *NIST/SEMATECH e–Handbook of Statistical Methods*, http://www.itl.nist.gov/div898/handbook/, March 3, 2021.

[3] Amos Gilat, *MATLAB, An Introduction with Applications, SI Version, 4ed*, Wiley, 2011, p. 77–78.

[4] www.mathworks.com/help/matlab/ref/randn.html.

[5] www.mathworks.com/help/matlab/math/controlling-random-number-generation.html.