Comput Stat (2013) 28:667–681
DOI 10.1007/s00180-012-0322-4

ORIGINAL PAPER

Efficient power computation for *r* out of *m* runs rules schemes

Rashid Mehmood · Muhammad Riaz · Ronald J. M. M. Does

Received: 5 April 2011 / Accepted: 13 March 2012 / Published online: 29 March 2012

Abstract In this article we develop a power computation code in the *R* language which provides an easy to use tool to researchers in designing Shewhart control charts. It enables researchers to use different existing and newly introduced sensitizing rules and runs rules schemes designed for Shewhart-type control charts for location and spread. The code provides researchers to compute the power for different options of *r* out of *m* rules/schemes. The code is flexible to apply for any sample size, false alarm rate, type of control limits (one- or two-sided), amount of shift in the process parameters and a variety of popular distributions for commonly used Shewhart-type control charts (i.e.$\bar{X}$, R, S and S^2 charts). These mentioned benefits of the developed functional code are only partially found in features of the existing software packages and these programs may be enhanced by adding the features of the developed code as a function in their libraries dealing with quality control charting.

Keywords Control charts · Efficient code · Normal and non-normal distributions · Power · *R* language · Runs rules

R. Mehmood · M. Riaz
Department of Statistics, Quaid-i-Azam University, Islamabad, Pakistan

R. Mehmood
Department of Commerce, Islamia University Bahawalpur Campus, Rahim Yar Khan, Pakistan

M. Riaz (✉)
Department of Mathematics and Statistics, King Fahd University of Petroleum and Minerals,
Dhahran 31261, Saudi Arabia
e-mail: riaz76qau@yahoo.com

R. J. M. M. Does
Department of Quantitative Economics, IBIS UvA, University of Amsterdam,
Plantage Muidergracht 12, 1018 TV Amsterdam, The Netherlands

1 Introduction

Processes show variations in its output. These variations are mainly of two types, namely natural and unnatural. The natural variations in a process (in-control process) are inherent. For the unnatural variations (out-of-control process) there exist special reasons which need to be identified. A timely identification of these special causes of variations helps boosting the quality of a process. In order to address and differentiate these two types of variations we have a very useful method, called statistical process control (SPC), which contains powerful tools like Pareto charts, check sheets, cause and effect diagrams, and control charts. In this SPC tool kit the control chart is the most important and commonly used tool due to its statistical framework. Shewhart-type control charts are very popular because of their simplicity. The charts are good in detecting large shifts in the process parameters. We may enhance their detection ability by supporting their design structures with some extra sensitizing rules and runs rules schemes.

Many authors have proposed runs rules schemes for this purpose; see for example Alwan (2000), Klein (2000), Khoo (2004), Koutras et al. (2007), Antzoulakos and Rakitzis (2008) and the references therein. Implementation of such schemes increases the sensitivity of the control chart for smaller shifts, but at the same time complicates their design structure. As a result performance evaluations of these types of charts become a difficult task. A popular performance evaluation criterion for comparing control charts is the power, which is the probability of declaring a process as out-of-control when it is actually out-of-control.

The most popular Shewhart-type control charts (i.e. $\bar{X}$, R, S and S^2 charts) generally work with only one sensitizing rule: a process is said to be out of control if the charting statistic exceeds one of the control limits (this is usually denoted by *1* out of *1* or (*1/1*)). This *1/1* is mainly sensitive for large shifts. By applying extra runs rules in the basic design structure of a control chart gives a push to their detection ability for smaller shifts. However, this will result in some complications. We list the following issues in this respect:

(i) difficulties in the power computation due to complicated design structures;
(ii) inflation in the false alarm rate in case of simultaneous application of more rules;
(iii) biasedness in case of separate use of each rule;
(iv) limited availability in software packages for power evaluation of different r/m rules;
(v) non-flexibility for other choices of false alarm rates.

In order to overcome these issues with the runs rules Riaz et al. (2011) provided an extended and compact form of the runs rules and recommended their application in industrial processes for efficient process monitoring.

There are software packages available which accommodate a few of the aforementioned runs rules or schemes and even these suffer some of the problems highlighted in (i)–(v) above. Therefore, some efficient code is needed which addresses and overcomes these complications and is flexible for a variety of distributions useful for practical purposes. In this study we develop an efficient code for different normal and non-normal distributions in the R language which addresses issues (i) through (v), particularly with reference to Riaz et al. (2011). The choice of the R language is made due to its friendly environment, easy access, interactive software environment for statistical computations and graphics, its de-facto standard behavior and implementation link with the S programming language, and its wide application among statisticians and researchers of other fields. Some more similar motivations may be found in Croux and Rousseeuw (1992), Nadarajah and Kotz (2007), Höhle (2007), Berens (2009), Harner et al. (2009), Hornik et al. (2009), Tierney (2009) and Spiring (2010).

Before discussing the said code in the R language we review first the methodological details of all the rules and procedures of Riaz et al. (2011).

We define a runs rules scheme as: "At least r out of m consecutive (r/m) points fall outside its respective signaling (control) limits $h_{r/m}$ which we may set on one side (lower or upper) or on both the sides of the sampling distribution of a control charting statistic. In this study we consider $m = 1, 2, 3, \ldots, 9$ and for each m we consider the choices $r = m - 2, m - 1, m$; where $1 \leq r \leq m$. In this way we have 24 possible rules of decision. In general terminology we refer these rules as r/m rules. In this set of rules the phrase "at least" is in fact the suggested modification for these runs rules schemes which helps to overcome the above mentioned problems (ii), (iii) and (v) and gives an attractive independence to each rule. The signaling limit $h_{r/m}$ is set according to a pre-specified value of false alarm rate for a given value of n and it is obtained using the mathematical expression $\alpha = \sum_{r<=m} \frac{m!}{r!(m-r)!} p^r (1 - p)^{(m-r)}$, where α is the pre-specified false alarm rate and p is the probability of a single point falling outside the respective signaling limits depending upon r and m. For more details the interested reader is referred to Riaz et al. (2011).

Different software packages, like MINITAB, SPSS, MATLAB, MAPLE, R, and S, do not have built-in functions for power computations of the different runs rules schemes on different types of control charts under a variety of distributions and also do not allow flexibility for fixing α at the desired level. An R package for statistical quality control namely "qcc" is given by Scrucca (2004) to monitor process characteristics. It is an efficient package for control charting of the charts like $\bar{X}$, R, S and S^2 charts, but it works with only one sensitizing rule (i.e. *1/1*) and misses the application of the other rules which are really needed to address the smaller shifts. The power computations accommodated by Scrucca (2004) (and similarly for many other packages e.g. MINITAB) are limited to very few rules/schemes (like *1/1, 2/3,* or 4/5) and with a limited scope (such as: not providing flexibility of picking any choice of 'r' and 'm'; not accommodative for choosing probability limits at any desired false alarm rate; not allowing to compute the detection power for any amount of shift in the parameter of interest), while the other rules/schemes, which may be more powerful at detecting different shifts, are missing in their environments. Therefore, we need an efficient code for power computations for all the rules in general, which may work for the commonly used distributional environments. Practitioners generally prefer statistical techniques (e.g. a control chart) which have a high power and they use it for their research proposals (cf. Mahoney and Magel 1996). So the power computation code of this study would be of great value for their future studies. Literature supporting the power evaluation criterion may be also found in Albers and Kallenberg (2006), Riaz (2008) and Montgomery (2009).

The functional code is accommodative for any sample size (n), false alarm rate (α), amount of shift in the parameters of interest, any type of control limits (one- or two-sided) for the design structures of $\bar{X}$, R, S and S^2 charts under a variety of distributional setups. The R code is written by using some built-in and some user-defined functions to program the aforesaid r/m runs rules schemes for all the mentioned choices of r and m to achieve the desired objectives. The code is flexible to cover different popular probability distributions of practical importance including Normal, t, Gamma, Chisquared, Weibull, Logistic, and Lognormal (cf. Schoonhoven and Does 2010; Abbasi and Miller 2011 and the references therein). It may be easily modified for other distributions as well. The code is available on the website: http://www.ibisuva.nl/en/Research/computer-codes.html

2 Description of the code

In this section we provide a detailed description of the code in terms of its development and functionality for the desired purposes. This description will help researchers to implement it easily for their desired use. The code mainly consists of three user-defined functions, namely "onepoint", "controllimit" and "power" to meet the desired objectives.

The function **"onepoint" (onepoint = function(alpha, r, m))** provides the probability of a single point falling outside the respective signaling limits. It enables the researcher to fix α at the desired level by applying the index search approach using the arguments "alpha", "*r*"and "*m*". The argument "alpha" is the pre-specified false alarm rate and it may take any value between 0 and 1, the argument "*r*" refers to the favorable points for an out-of-control signal and the argument "*m*" refers to the total consecutive points considered in a given rule.

The function **"controllimit"(controllimit = function(chart,r,m,side,n,simu, mean,sd,alpha, dist))** results in the signaling (control) limit(s) for a given r/m rule. It depends upon the arguments "chart", "*r*", "*m*" "side", "n", "simu", "mean", "sd", "alpha" and "dist". The argument "chart" can take one of the four possible choices, namely "xbar", "S", "R", "S2". The argument "side" relates to the nature of control limits and it may take one of the three possible options namely "L" for lower sided, "U" for upper sided and "T" for two sided limits. The argument "n" is the sample size and it may take any value. The argument "simu" shows the number of simulations to be used to compute the desired power of a control chart. The argument "dist" stands for the particular probability distribution under investigation. The arguments "mean" and "sd" represent the specified values of standard deviation and mean respectively of the normal distribution (which is the default distribution in our code and the default values for "mean" and "sd" are taken as 0 and 1 respectively). The other arguments are as defined earlier.

The function "power"

$$\begin{aligned}\mathbf{power} = \mathbf{function}(&\mathbf{chart}, \mathbf{r}, \mathbf{m}, \mathbf{side}, \mathbf{n}, \mathbf{simu}, \mathbf{mean}, \mathbf{sd}, \mathbf{alpha}, \mathbf{delta}, \mathbf{dist},\\ &\mathbf{df}, \mathbf{shape}, \mathbf{scale}, \mathbf{meanlog}, \mathbf{sdlog}, \mathbf{location})\end{aligned}$$

is the front end function for the researcher. The arguments used in this function are “chart”, “r”, “m”, “side”, “n”, “simu”, “mean”, “sd”, “alpha”, “delta”, “dist”, “df”, “shape”, “scale”, “meanlog”, “sdlog” and “location”. The argument “delta” represents the amount of shifts in terms of sigma units (i.e. delta*sd) which we want to detect for a given chart. In the case of the $\bar{X}$ chart, delta = 0 refers to an in-control situation and otherwise out-of-control. For the other three charts delta = 1 refers to an in-control situation and otherwise out-of-control. The argument “dist” represents the particular probability distribution we are concerned and once it is entered as an argument then its corresponding parameter values need to be specified as follow up arguments. The following table may be used to enter the arguments about distributions and their corresponding parameters. By entering the corresponding parameters of a specific distribution the other parameter related arguments automatically become null. The arguments “mean”, “sd” and “dist” by default are set as zero, one and normal respectively in the power function because of its popularity in literature. All the other arguments used in **“power”** function are as defined earlier and it calls the function “controllimit” to use the control limits.

Arguments for the distributions along with their corresponding parameters used in power function

No	Distribution	Argument for dist	Parameter(s) of dist
1	Normal	“normal”	mean, sd
2	T	“t”	df
3	Chi-squared	“chisquared”	df
4	Weibull	“weibull”	shape, scale
5	Logistic	“logistic”	location, scale
6	Gamma	“gamma”	shape, scale
7	Log-normal	“lognormal”	meanlog, sdlog

The **“power”** function obtains the corresponding statistic values according to the choice of “chart” and then these statistics values are consecutively compared with the values of the control limit obtained from the function “controllimit”. The number of times a point goes out of control is counted. The function “power” ends with an output command (statement) in its coded version which provides the summary of the final output of the desired control chart in the form of probability of detecting out-of-control signals (i.e. power). A summary of the output is shown in the *R* editor or console in an attractive format for the user. The order in which the code should be carried out is **“onepoint” -> “controllimit” -> “power”**.

3 Execution procedure of the code

In this section we list the necessary steps which help the users and researchers to adopt our developed code for computing the control limits of a control chart for a given sample size and finally evaluating the detection (power) ability for different probability distributions.

(i) Run the code (as provided on the web link http://www.ibisuva.nl/en/Research/computer-codes.html) in the *R* editor or console and save the workspace so that it may be loaded before using the **“power”** function next time (or alternatively

"power" may be added in the *R* library as a function of quality control package).

(ii) To get the final output in the form of probability distribution, control chart, control limit, rule type, false alarm rate, delta and power, execute the following function **"power"** after giving the desired options/values to the arguments used in it e.g. **power(chart = "xbar",r = 1,m = 1,side = "T",n = 5,simu = 10000, alpha = 0.0027**, **delta = 0, dist = "cauchy",location = 10,scale = 10)** is for the $\bar{X}$ chart using the *1/1* rule at $\alpha = 0.0027$ using two sided control limits for sample size n = 5 and an in-control situation (delta = 0). The other options for these arguments may easily be entered following the description of Sect. 2 as per requirements.

(iii) After the execution of the **"power"** function we will get the final output of the data which researchers and users may want in the form of summary display in the R-console providing the information of distribution (normal, t, chisquared, logistic, weibull, lognormal, gamma), control chart type, control limits type (one- or two-sided), runs rules type (r/m), false alarm rate, delta and power. This summary in the display is easy to understand and is very attractive for researchers to quickly have an idea of the chart's ability.

4 Illustrations and demonstrations of the code's results

To exemplify the application of our developed code for the said purposes we apply our code for a few of the runs rules schemes with different distributions at different false alarm rates using some choices of delta (shifts) for the control charts considered in this study. Also we show what kind of output is obtained by our code. These results may be compared with those cases for which theoretical results are also available for the validation of our power computation code.

For the *1/1* rule on the $\bar{X}$ chart we upload the code (as provided on the web link http://www.ibisuva.nl/en/Research/computer-codes.html) in the *R* console and execute the following statement:

$$\mathbf{power}\left(\mathbf{chart} = \text{``}\mathbf{xbar}'', \mathbf{r} = \mathbf{1}, \mathbf{m} = \mathbf{1}, \mathbf{side} = \text{``}\mathbf{T}'', \mathbf{n} = \mathbf{5}, \mathbf{simu} = \mathbf{10000}, \\ \mathbf{alpha} = \mathbf{0.0027}, \mathbf{delta} = \mathbf{0}\right)$$

which results in the following output in the *R* console:

Summary for out-of-control probability

Distribution:	normal
Control Chart:	xbar
Control Limit (T):	1.341630–1.341630
Rule Type:	1 / 1
False Alarm Rate:	0.0027
Sample Size:	5
Delta:	0
Power:	0.0027

We can see in the above summary that the code has given the same false alarm rate as desired and the value of mean, sd and dist are not provided in power function as it the default case of our code as mentioned in Sect. 2.

As another example for the 2/*3* rule on the *R* chart using t distribution (with 5 degress of freedom) we upload the same code and execute the following statement:

```
power(chart="R",r=2,m=3,side="T",n=5,simu=100000,alpha=0.0027,delta=2,
  dist="t",df=5)
```

which results in the following output in the *R* console:

Summary for out-of-control probability

Distribution:	t
Control Chart:	R
Control Limit (T):	6.976905 0.8110547
Rule Type:	2 / 3
False Alarm Rate:	0.0027
Sample Size:	5
Delta:	2
Power:	0.1793

For some more choices of sample sizes, different control charts, varying runs rules schemes, different amounts of shifts at a given false alarm rate, and varying distribution. we also have run our power code and got the following results (which are self explanatory):

Summary for out-of-control probability

Distribution:	t
Control Chart:	xbar
Control Limit (U):	2.012745
Rule Type:	1 / 2
False Alarm Rate:	0.0027
Sample Size:	5
Delta:	2
Power:	0.7414

Summary for out-of-control probability

Distribution:	weibull
Control Chart:	xbar
Control Limit (U):	1.085408
Rule Type:	1 / 2
False Alarm Rate:	0.0027
Sample Size:	5
Delta:	2
Power:	1

Summary for out-of-control probability

Distribution:	logistic
Control Chart:	S2
Control Limit (L):	0.0676655
Rule Type:	1 / 2
False Alarm Rate:	0.0027
Sample Size:	5
Delta:	0.1
Power:	0.1663

Summary for out-of-control probability

Distribution:	lognormal
Control Chart:	S2
Control Limit (L):	0.2564418
Rule Type:	1 / 2
False Alarm Rate:	0.0027
Sample Size:	7
Delta:	0.01
Power:	0.9236

Summary for out-of-control probability

Distribution:	chisquared
Control Chart:	S
Control Limit (T):	6.864225 0.736658
Rule Type:	2 / 4
False Alarm Rate:	0.0027
Size:	5
Delta:	0.1
Power:	1

Summary for out-of-control probability

Distribution:	gamma
Control Chart:	xbar
Control Limit (T):	155.0144 145.0091
Rule Type:	7 / 9
False Alarm Rate:	0.0027
Sample Size:	7
Delta:	2
Power:	0.0068

Similarly, any option may be entered in the **“power”** function to compute the out-of-control probability (i.e. power). It is to be noted that the accuracy level may be increased by increasing the number of simulations.

From the above results we see that using our function **“power”** it is quite easy to calculate the power (along with the respective control limits) for the $\bar{X}$, R, S and S^2 control charts with any runs rule scheme under consideration at any false alarm rate and for any given amount of shift in the process parameters for a variety of probability distributions, which is in general not the case with the built-in functions of other software packages. It may also be noted that for those cases where theoretical (or otherwise) results for the power of a control chart are available, the developed function **“power”** has given almost the same results which ensures the validity of our function for the said computational purposes (for example see and compare the results of $\bar{X}$, R, S and S^2 charts given by Alwan (2000), Klein (2000), Khoo (2004), Koutras et al. (2007), Antzoulakos and Rakitzis (2008), Riaz (2008) and Montgomery (2009).

For the sake of further illustration and comparisons, we have taken different choices of the arguments for different charts and distributions to create some selective tables which are provided in the “Appendix” (cf. Tables 1, 2, 3, 4, 5, 6, 7 in the Appendix). We have also analyzed the results of these tables to furnish some comparisons among different methods under consideration.

The output of these tables (cf. Tables 1, 2, 3, 4, 5, 6, 7) indicated that: (i) the code is capable to work efficiently for different charts under under normal and non-normal

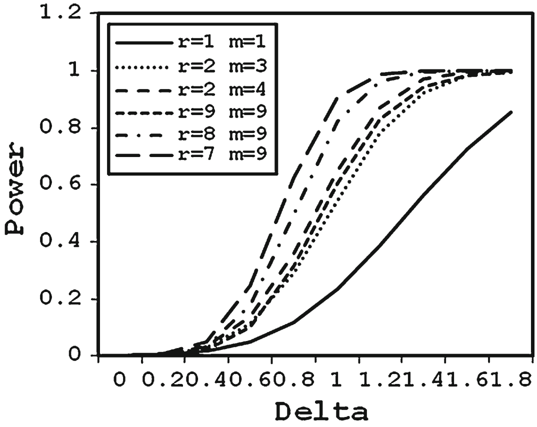


Fig. 1 dist = “normal”, chart = “xbar”, n = 5, alpha = 0.0027, mean = 0, sd = 1, side = “T”

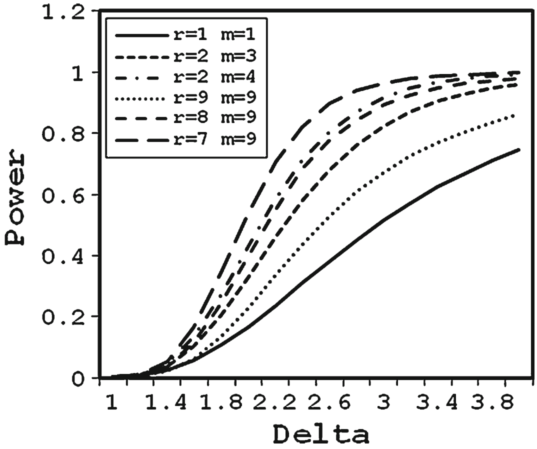


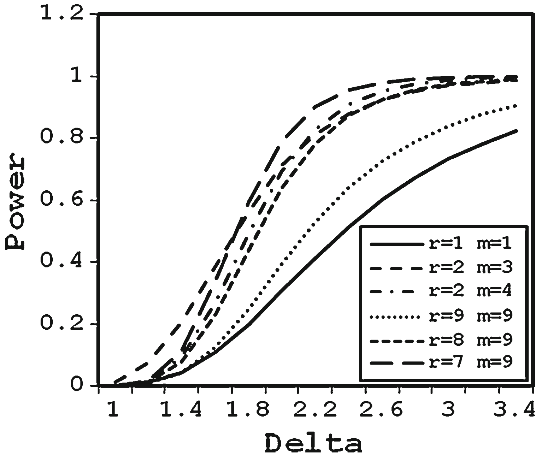
Fig. 2 dist = “logistic”, chart = “R”, n = 5, alpha = 0.0027, location = 2, scale = 1, side = “T”

distrubutional environments; (ii) the coded runs rules schemes enhance the detection ability of all the charts under different distributional setups (cf. Figs. 1, 2, 3); (iii) in general a higher choice of “m” results into more efficiency and for a specific choice of “m” a superior performance is experienced when “r” is chosen as “$m - 2$” followed by “$m - 1$” and finally “m” for $\bar{X}$, R, S charts under different distributional setups (cf. Figs. 1, 2, 3).

5 Summary, conclusions, recommendations and future proposals

It has never been an easy task to fix the false alarm rate at any level with different runs rules schemes to compute the power of a control chart for a specific probability distribution. The existing software packages generally work with a few sensitizing rules/runs rules schemes for very few distributional environments of practical importance and at very limited choices of false alarm rates. We have developed a functional code which can handle this for many rules or schemes and for any choice of the false

Fig. 3 dist = "gamma", chart = "S", n = 5, alpha = 0.0027, shape = 10, scale = 1, side = "T"



alarm rate under a variety of commonly used distributions (this we have done for the $\bar{X}$, R, S and S^2 charts at the moment and it may be done for other charts as well). The code provides an attractive summary display of the power along with the control limits of the corresponding control charts at any false alarm rate and for any amount of shift in the R console directly for different distributions including Normal, t, Gamma, Chisquared, Weibull, Logistic, and Lognormal

This code may be easily modified for many other distributions other than those considered in this article. Moreover, it may be extended for the EWMA and CUSUM charts. Also the code may be modified to obtain the average run length (ARL) of the control charts. Further, the code may be enhanced to accommodate a list of values for delta simultaneously and then display the power/ARL results in the form of tables or graphical displays, i.e. their corresponding curves.

Keeping in mind the above mentioned features of the proposed functional code we suggest its inclusion in the library of the R language as a quality control tool/package with the name **"power"** just like the **"qcc"** package of Scrucca (2004). In general, it may be considered as a contribution for any package in the form of functional code for the said purposes. The application of this code may be particularly of more benefit for the researchers who rely on the power criterion for the selection of a statistical procedure for their research proposals and hence may use this functional code in their future studies (cf. Mahoney and Magel 1996).

Acknowledgments The authors are thankful to the reviewer(s) and editor for the useful comments to improve the initial version of the article. The author Muhammad Riaz is indebted to King Fahd University of Petroleum and Minerals Dhahran Saudi Arabia for providing excellent research facilities through project SB111008. The authors are also grateful to Dr. Zawar Hussain and Mr. Muhammad Kashif Faiz for their helpful comments on some aspects of the code presentation.

Appendix

See Tables 1, 2, 3, 4, 5, 6 and 7.

Table 1 dist = “normal”, chart = “xbar”, n = 5, alpha = 0.0027, mean = 0, sd = 1, side = “T”

Delta	r = 1, m = 1	r = 2, m = 3	r = 2, m = 4	r = 9, m = 9	r = 8, m = 9	r = 7, m = 9
0	0.0027	0.0027	0.0027	0.0027	0.0027	0.0027
0.2	0.006	0.0064	0.0068	0.005	0.0058	0.0073
0.4	0.0188	0.03	0.0347	0.0228	0.0348	0.0482
0.6	0.0513	0.1128	0.1367	0.1015	0.1781	0.2484
0.8	0.1191	0.2898	0.3604	0.3148	0.4965	0.6264
1	0.2342	0.546	0.6478	0.6043	0.8214	0.9093
1.2	0.3876	0.7823	0.8721	0.8323	0.9622	0.9893
1.4	0.5655	0.9251	0.9705	0.9439	0.9952	0.9994
1.6	0.7272	0.9829	0.9963	0.9847	0.9996	1
1.8	0.8539	0.9972	0.9997	0.9966	1	1
2	0.9343	0.9997	1	0.999	1	1
2.2	0.9751	1	1	0.9998	1	1
2.4	0.9919	1	1	1	1	1
2.6	0.9979	1	1	1	1	1
2.8	0.9995	1	1	1	1	1
3	0.9999	1	1	1	1	1

Table 2 dist = “weibull”, chart = “S2”, n = 5, alpha = 0.0027, shape = 10, scale = 1, side = “T”

Delta	r = 1, m = 1	r = 2, m = 3	r = 2, m = 4	r = 9, m = 9	r = 8, m = 9	r = 7, m = 9
1	0.0027	0.0027	0.0027	0.0027	0.0027	0.0027
1.2	0.0051	0.0054	0.006	0.0045	0.005	0.0053
1.4	0.0104	0.0117	0.014	0.008	0.0103	0.0122
1.6	0.0186	0.0227	0.0282	0.0141	0.0204	0.026
1.8	0.0299	0.0399	0.051	0.0231	0.0371	0.0497
2	0.0442	0.0631	0.084	0.0357	0.0608	0.0859
2.2	0.0602	0.0911	0.123	0.0514	0.0911	0.1325
2.4	0.0778	0.1247	0.1674	0.071	0.1292	0.1866
2.6	0.0982	0.1616	0.2185	0.0941	0.1703	0.2466
2.8	0.1197	0.2034	0.2715	0.1192	0.2173	0.3112
3	0.1414	0.2444	0.3281	0.1461	0.2669	0.3734
3.2	0.1637	0.2883	0.3821	0.1743	0.315	0.4387
3.4	0.1855	0.3299	0.435	0.2021	0.3637	0.4976
3.6	0.2085	0.372	0.486	0.2325	0.4101	0.5533
3.8	0.2312	0.4121	0.533	0.2606	0.4559	0.6057
4	0.2533	0.4506	0.5779	0.2897	0.4979	0.6517
4.2	0.2764	0.4871	0.6194	0.3163	0.539	0.6919
4.4	0.2987	0.5222	0.6557	0.3426	0.5749	0.7295

Table 3 dist = “logistic”, chart = “R”, n = 5, alpha = 0.0027, location = 2, scale = 1, side = “T”

Delta	r = 1, m = 1	r = 2, m = 3	r = 2, m = 4	r = 9, m = 9	r = 8, m = 9	r = 7, m = 9
1	0.0027	0.0027	0.0027	0.0027	0.0027	0.0027
1.2	0.0083	0.0099	0.0116	0.0072	0.009	0.0109
1.4	0.0257	0.0386	0.0487	0.0241	0.0379	0.0529
1.6	0.0578	0.104	0.1373	0.0634	0.1141	0.1681
1.8	0.1066	0.2049	0.273	0.1355	0.2454	0.3454
2	0.168	0.33	0.4302	0.2293	0.4017	0.5405
2.2	0.2365	0.4623	0.5852	0.3365	0.5528	0.7056
2.4	0.3095	0.5795	0.7118	0.4346	0.6835	0.8212
2.6	0.3804	0.6795	0.8056	0.5284	0.7764	0.8971
2.8	0.4506	0.7602	0.8726	0.6099	0.8454	0.9397
3	0.5149	0.8231	0.9181	0.6731	0.8956	0.9654
3.2	0.572	0.87	0.9479	0.7287	0.9272	0.9808
3.4	0.6239	0.9041	0.9668	0.7692	0.9488	0.9888
3.6	0.6694	0.9297	0.9789	0.8049	0.9638	0.9932
3.8	0.7105	0.9487	0.9866	0.8365	0.9735	0.996
4	0.7466	0.9621	0.9916	0.8621	0.9814	0.9975
4.2	0.7775	0.9712	0.9945	0.882	0.9864	0.9984
4.4	0.8044	0.9785	0.9963	0.8989	0.9901	0.999

Table 4 dist = “t”, chart = “xbar”, n = 5, alpha = 0.0027, df = 5, side = “T”

Delta	r = 1, m = 1	r = 2, m = 3	r = 2, m = 4	r = 9, m = 9	r = 8, m = 9	r = 7, m = 9
0	0.0027	0.0027	0.0027	0.0027	0.0027	0.0027
0.2	0.0034	0.0042	0.0043	0.0045	0.005	0.0054
0.4	0.0054	0.0116	0.012	0.0148	0.0204	0.0249
0.6	0.0102	0.0354	0.0394	0.0552	0.0874	0.1157
0.8	0.0206	0.0978	0.1111	0.1619	0.2714	0.351
1	0.0404	0.2237	0.2601	0.3545	0.5528	0.6681
1.2	0.0783	0.4178	0.4867	0.5705	0.7945	0.8896
1.4	0.1414	0.6334	0.7212	0.7499	0.9268	0.9745
1.6	0.2358	0.8132	0.887	0.8665	0.9791	0.9956
1.8	0.362	0.9232	0.966	0.9338	0.9944	0.9994
2	0.5068	0.9735	0.9926	0.9664	0.9986	0.9999
2.2	0.6494	0.9921	0.9987	0.9825	0.9997	1
2.4	0.7721	0.9979	0.9998	0.9899	0.9999	1
2.6	0.8646	0.9995	1	0.9946	1	1
2.8	0.9244	0.9999	1	0.9975	1	1
3	0.9602	1	1	0.999	1	1
3.2	0.9801	1	1	0.9993	1	1
3.4	0.9902	1	1	0.9996	1	1

Table 5 dist = "lognormal", chart = "S2", n = 5, alpha = 0.0027, meanlog = 1, sdlog = 1, side = "T"

Delta	r = 1, m = 1	r = 2, m = 3	r = 2, m = 4	r = 9, m = 9	r = 8, m = 9	r = 7, m = 9
1	0.0027	0.0027	0.0027	0.0027	0.0027	0.0027
1.2	0.0028	0.0028	0.0027	0.0027	0.0028	0.0027
1.4	0.0031	0.0031	0.0033	0.003	0.0031	0.0032
1.6	0.0035	0.0037	0.0038	0.0035	0.0038	0.0039
1.8	0.0042	0.0046	0.0046	0.0042	0.0047	0.005
2	0.0049	0.0055	0.0057	0.0051	0.0058	0.0064
2.2	0.0055	0.0066	0.0069	0.0062	0.0073	0.0082
2.4	0.0062	0.0079	0.0084	0.0075	0.0093	0.0104
2.6	0.007	0.0093	0.0099	0.0088	0.0114	0.0133
2.8	0.0076	0.0109	0.0117	0.0105	0.0139	0.0163
3	0.0086	0.0128	0.0137	0.0124	0.017	0.0201
3.2	0.0095	0.0147	0.0157	0.0144	0.0202	0.0246
3.4	0.0103	0.0168	0.0181	0.0168	0.0241	0.0298
3.6	0.011	0.0188	0.0205	0.0194	0.0283	0.0352
3.8	0.0118	0.0211	0.0233	0.0222	0.0329	0.041
4	0.0128	0.0233	0.0262	0.0252	0.0378	0.0479
4.2	0.0137	0.0257	0.0292	0.0286	0.0431	0.0553
4.4	0.0146	0.0281	0.0323	0.0323	0.0489	0.0631

Table 6 dist = "gamma", chart = "S", n = 5, alpha = 0.0027, shape = 10, scale = 1, side = "T"

Delta	r = 1, m = 1	r = 2, m = 3	r = 2, m = 4	r = 9, m = 9	r = 8, m = 9	r = 7, m = 9
1	0.0027	0.0152	0.0027	0.0027	0.0027	0.0027
1.2	0.0124	0.0742	0.0177	0.0103	0.0139	0.0178
1.4	0.0454	0.2065	0.0945	0.0448	0.0781	0.113
1.6	0.1097	0.3868	0.2707	0.1249	0.2319	0.3396
1.8	0.2013	0.5661	0.4947	0.2508	0.4411	0.5961
2	0.3067	0.7109	0.693	0.3959	0.6376	0.7906
2.2	0.4143	0.8118	0.8307	0.5302	0.7829	0.902
2.4	0.5145	0.8803	0.9097	0.6425	0.8746	0.9563
2.6	0.6029	0.9254	0.9536	0.726	0.9275	0.9813
2.8	0.6743	0.9533	0.9767	0.7912	0.9578	0.9919
3	0.7344	0.9711	0.9883	0.8417	0.9756	0.9963
3.2	0.7834	0.9818	0.9942	0.8785	0.9857	0.9983
3.4	0.8243	0.9885	0.997	0.9063	0.9915	0.9992
3.6	0.8564	0.9926	0.9985	0.9264	0.9949	0.9996
3.8	0.883	0.9951	0.9992	0.9429	0.9967	0.9998
4	0.9038	0.9967	0.9996	0.9549	0.998	0.9999
4.2	0.9209	0.9978	0.9998	0.9643	0.9987	1

Table 7 dist = "chisquared", chart = "R", n = 5, alpha = 0.0027, df = 5, side = "T"

Delta	r = 1, m = 1	r = 2, m = 3	r = 2, m = 4	r = 9, m = 9	r = 8, m = 9	r = 7, m = 9
1	0.0027	0.0027	0.0027	0.0027	0.0027	0.0027
1.2	0.0066	0.0086	0.0097	0.0063	0.0076	0.0088
1.4	0.0178	0.0299	0.0365	0.0193	0.0291	0.0384
1.6	0.0409	0.078	0.1006	0.0497	0.085	0.1232
1.8	0.0749	0.1558	0.2045	0.1052	0.1862	0.2679
2	0.1192	0.2588	0.3363	0.1815	0.3237	0.4435
2.2	0.1715	0.3726	0.4757	0.2739	0.4638	0.6124
2.4	0.2293	0.4859	0.6056	0.3663	0.5965	0.7438
2.6	0.2898	0.5872	0.7132	0.4569	0.7035	0.8394
2.8	0.3505	0.6744	0.7957	0.542	0.7882	0.9008
3	0.4101	0.7464	0.8581	0.6161	0.8487	0.9397
3.2	0.467	0.8053	0.9028	0.6817	0.8948	0.9637
3.4	0.5194	0.8508	0.9343	0.731	0.9262	0.9789
3.6	0.5675	0.8854	0.9554	0.7752	0.9483	0.9875
3.8	0.6112	0.9131	0.9697	0.8089	0.9629	0.9926
4	0.6509	0.9334	0.9798	0.8374	0.9735	0.9955
4.2	0.6875	0.9486	0.9862	0.865	0.9808	0.9972
4.4	0.7199	0.9607	0.9905	0.8882	0.9859	0.9982

References

Abbasi SA, Miller A (2011) On proper choice of variability control chart for normal and non-normal processes. Qual Reliab Eng Int. doi:10.1002/qre.1244

Albers W, Kallenberg WCM (2006) Self-adapting control charts. Statistica Neerlandica 60:292–308

Alwan LC (2000) Statistical process analysis. McGraw-Hill International Editions, Singapore

Antzoulakos DL, Rakitzis AC (2008) The modified r out of m control chart. Commun Stat Simul Comput 37:396–408

Berens P (2009) CircStat: a MATLAB toolbox for circular statistics. J Stat Softw 31(10):1–21

Croux C, Rousseeuw PJ (1992) Time efficient algorithms for two highly robust estimators of scale. In: Dodge Y, Whittaker J (eds), Computational Statistics, vol 1, pp 411–428. Physika-Verlag, Heidelberg

Harner EJ, Luo D, Tan J (2009) JavaStat: a Java/R-based statistical computing environment. Comput Stat 24:295–302

Höhle M (2007) Surveillance: an R package for the monitoring of infectious diseases. Comput Stat 22: 571–582

Hornik K, Buchta C, Zeileis A (2009) Open-source machine learning: R meets Weka. Comput Stat 24: 225–232

Khoo MBC (2004) Design of runs rules schemes. Qual Eng 16:27–43

Klein M (2000) Two alternatives to the Shewhart X control chart. J Qual Technol 32:427–431

Koutras MV, Bersimis S, Maravelakis PE (2007) Statistical process control using Shewhart control chart with supplementary runs rules. Methodol Comput Appl Probab 9:207–224

Mahoney M, Magel R (1996) Estimation of the power of the Kruskal-Wallis test. Biometrical J 38:613–630

Montgomery DC (2009) Introduction to statistical quality control, 6th edn. Wiley, New York

Nadarajah S, Kotz S (2007) Programs in R for computing truncated *t* distributions. Qual Reliab Eng Int 23:273–278

Riaz M (2008) Improved and robust monitoring in statistical process control. PhD thesis, University of Amsterdam

Riaz M, Mehmood R, Does RJMM (2011) On the performance of different control charting rules. Qual Reliab Eng Int 27(8):1059–1067

Schoonhoven M, Does RJMM (2010) The $\bar{X}$ control chart under non-normality. Qual Reliab Eng Int 26:167–176

Scrucca L (2004) qcc: an R package for quality control charting and statistical process control. R News 4/1:11–17

Spiring F (2010) Exploring process capability with Mathematica. Qual Reliab Eng Int. Available online with doi:10.1002/qre.1112

Tierney L (2009) Code analysis and parallelizing vector operations in R. Comput Stat 24:217–223