On Efficient Computation and Implementation of r/m Run Rules Schemes in Statistical Quality Control

BY

Rashid Mehmood

DEPARTMENT OF STATISTICS
QUAID-I-AZAM UNIVERSITY
ISLAMABAD, PAKISTAN

2010

بسم الله الرحمن الرحيم

In the name of ALLAH,
The Most Beneficent,
The Most Merciful.

On Efficient Computation And Implementation of r/m Run Rules Schemes In Statistical Quality Control

BY

Rashid Mehmood

A THESIS SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MSc. IN STATISTICS

DEPARTMENT OF STATISTICS
QUAID-I-AZAM UNIVERSITY
ISLAMABAD, PAKISTAN

2010

Certificate

On Efficient Computation and Implementation of r/m Run Rules Schemes inStatistical Quality Control

BY

Rashid Mehmood

A THESIS SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MSc.

IN STATISTICS

We accept this thesis as conforming to the required standards.

1. ____________ 8/11/2010

Dr. Javid Shabbir
(Chairman)

2. ____________

Dr. Muhammad Riaz
(Supervisor) (on leave)

3. ____________

(External Examiner)

Date: ____________

DEPARTMENT OF STATISTICS
QUAID-I-AZAM UNIVERSITY
ISLAMABAD, PAKISTAN
2010

DECLARATION

I hereby solemnly declare that the thesis entitled "On Efficient Computation and Implementation of r/m Run Rules Schemes in Statistical Quality Control" submitted by me for the partial fulfillment of the Master degree in Statistics, is the original one and has not been submitted concurrently or latterly to this or any other university for any other degree.

Date: ______________ Signature: ______________

(Rashid Mehmood)

Dedicated to

"My Dear Parents "

"Please always know I love you and no one can
Take your place. Years may come and go but your memory will never be erased"

Acknowledgements

All praise is to ALMIGHTY ALLAH, creator of the heavens and earths and Lord of lords, it is by His blessing, who gave me potential and ability to complete this dissertation. All of my respect goes to HOLY PROPHET MUHAMMAD(PBUH) who emphasized the significance of knowledge and research.

First of all, I would like to pay tributes and thanks to my parents for my support and love, unending and countless prayers buoyed me up and moral support from my brothers and sister.I am very thankful to my supervisor Dr. Muhammad Riaz whose attention has always been a great source of inspiration in my life and whose sustained hope in me has led me to where I stand today.I offer tribute to Dr. Javid shabir Chairmen , Department Of Statistics for his affectionate attitude and overall patronization during my stay in the department. I am also grateful to Dr.Zawar Hussian , Dr.Zahid Asghar, Ms .Maryam for enlightening my statistical knowledge during the course work.

Special thanks are extended to all my friends, Zulqarnain Haider, Muhammad Bilal , Zulkaif umer shah, Rafaqat Malik, Rizwan Iqbal , Shazad Munir, Faisal Altaf, Ijaz Shah, Muhammad Abid, Dr Wajid, Raja Fawad Zafar, Muhammad Imran, and all those students residence in hostel.

Rashid Mehmood

Table of contents

Chapter 1

Introduction

Quality control is very important for manufacturing concerns because every company wishes its process to remain in control. But these processes are affected by natural and unnatural factors which result into variations which are generally categorized into two types namely natural and unnatural. Natural variation is permanent, small in magnitude and not easy to remove, e.g. customer satisfaction level, rain, etc. On the other hand unnatural variation is controllable, large in magnitude and easy to remove, e.g. batch of raw material having low quality will disturb the quality of produced items. But it can be controlled by changing the new batch of raw material. To check these variations Statistical Process Control (SPC) is there to serve the purpose and it consists of a tool kit mainly containing seven tools: Flowchart, Check Sheet, Pareto Diagram, Histogram, Cause & Effect Diagram, Scatter Diagram and Control Chart. Among these tools some are used for preliminary purpose to search the assignable causes (e.g. Flowchart, Pareto diagram, and Cause and Effect Diagram). Some of these tools are used for gathering information (e.g. Check Sheets). The tools like Histogram, Scatter Diagram, and Control Chart are used for information display.

In the SPC tool kit control charts are the most widely used tools which provide graphical display of quality characteristic(s). The structure of a control chart generally consists of centerline (CL), upper control limit (UCL), and lower control limit (LCL) and the process is said to be statistically in-control if behavior of the process characteristic(s) remains within these control limits. There are two types of control charts namely

'Memory Control Charts' and 'Memory Less Control Charts'. The Memory Control Charts provide the information about the current and relates the current pattern of process with the past. On the other hand Memory Less Control Charts provide information only about the current status of the process. The Memory Less Control Charts are also called Shewart's type control charts.

In Shewart's control charting procedures we have two further types namely variable and attribute control charts. Variable control chart is used for measurable data e.g. weight of an items generated through machine with regular interval of time whereas attribute control is used for categorical data e.g. number of conforming or non-conforming items produced by a machine with regular intervals of time. The most common Shewart's variable control charts include $\bar{X}$, S, S^2, R while in attributes type charts p, c, np, u are the most common choices.

The focus of this study will be Shewart's variable control charts particularly $\bar{X}$, S, S^2, R charts. In these charts a single point outside the control limits show the special cause of variation in the process and it behaves efficiently only for larger shifts. To detect such situation(s) different runs rules/schemes are developed by a number of researchers to make the design structures of these Shewart's type control charts (like $\bar{X}$, S, S^2, R charts) sensitive for smaller shifts as well These rules/schemes (known as runs or patterns) are used to check or test overall randomness in the data series and to search out short sequence of observations embedded within the overall data series which are inconsistent with randomness. These extra sensitizing runs rules schemes have their merits but have some serious issues which need to addressed very carefully. These problems mainly include: biasedness and non-monotonicity in case of separate use of each rule in general and hence no independent identity of any rule for different types of shifts; need of simultaneous application of more rules at a time which causes an inflated false alarm rate and unattractive structures for practitioners use; limited availability of the softwraes/packages which are capable to accommodate all these runs rules schemes and provide flexibility for any false alarm rate. To address these issues we have planned this study.

The organization of the thesis is as: Chapter 2 will highlight some issues with the runs rules schemes and provide some redefining mechanism to give an independent identity to each scheme and take care of the biasedness and non-monotonicity issues so that simultaneous application of more than one rules at a time may be avoided which would help making the design structures of control charts more attractive. In Chapter 3 we will handle the limitation issue of softwares/packages to accommodate very few rules for power computation and develop a code in $\mathbb{R}$ language to facilitate the power computation of different rules for any false alarm rate which is not an easy task in general using different softwares/packages. In Chapter 4 we will extend the use of our $\mathbb{R}$ code to provide an ease to practitioners to implement these different runs rules schemes easily. Finally Chapter 5 will summarize the findings of our study.

Chapter 2

On the Performance of Different Control Charting Rules

There are a number of control charting rules used with different control charts to decide between the two states of control namely in-control and out-of-control. Some issues with these rules will be highlighted in this chapter. By redefining and listing a set of rules we will evaluate their performance on the S^2, $\bar{X}$, R and S charts. Also we will compare the performance of these rules using their power curves to figure out the superior ones. Application of few of these rules with the real datasets will be also shown to highlight their detection ability and use for practitioners.

1. Introduction

It is quite common to use different sensitizing rules with the design structures of Shewhart's type control charts so that smaller shifts may also be addressed along with the larger ones. The cost we have to pay for this gain is the increase in false alarm rate. There is another serious issue with these sensitizing rules and that is these rules are not able to work independently to address different magnitudes of shifts. For example we list here three sensitizing rules from Alwan (2000):

***Rule# i*:** Signal out of control if a single point falls beyond zone A (symbolically this rule would be denoted by 1/1).

***Rule# ii*:** Signal out of control if 2 of 3 successive points on one side of the center line fall in zone A or beyond (symbolically this rule would be denoted by 2/3).

Rule# iii: Signal out of control if 4 of 5 successive points on one side of the center line fall in zone B or beyond (symbolically this rule would be denoted by 4/5).

where zones A and B mentioned in the above three rules may be obtained using 3-sigma and 2-sigma limits respectively or using the probability limits approach to get the pre-specified false alarm rate (α). More details about these zones may be seen in Alwan (2000).

In the abovementioned rules, ***Rule# i*** can address larger shifts whereas ***Rules# ii*** and ***iii*** are good at detecting smaller shifts. To simultaneously address both types of shifts we have to combine these rules (e.g. Rule# *ii* with Rule# *i* and Rule# *iii* with Rule# *i*) or some other similar rules with a control chart structure. Imposing more rules simultaneously complicates the application of control chart and also inflates the false alarm rate. The problem of inflated false alarm rate may be handled by appropriately adjusting the control limits coefficient used with each rule as is done by Klien (2000) and Khoo (2004). To overcome the complications of applying more rules at a time we suggest a separate use of each rule. An independent and separate application of these rules and similar other rules with a control chart structure may cause an abnormal power patterns.

To illustrate this problem more specifically we consider the upper sided S^2 chart. We have obtained the control lines (for zones A and B) of ***Rules# i, ii* and *iii*** in the forms of h_0, h_{00} and h_{000} respectively to get a false alarm rate (α) equal to 0.0027 for the S^2 chart (upper sided). Now out-of-control signal is given by ***Rule# i*** if 1 point falls out side h_0, by ***Rule# ii*** if 2 out of 3 consecutive points fall between h_{00} & h_0 and by ***Rule# iii*** if 4out of 5 consecutive points fall between h_{000} & h_{00}. For this set up, discriminatory powers are computed of the above mentioned three rules assuming that samples are coming from $N(\mu, \delta\sigma)$ where δ represents the amount of shift. Here $\delta = 1$ implies that there is no shift in σ (hence process is in-control) and $\delta > 1$ means that there is an increase in σ (hence process is out-of-control). The results of discriminatory powers at $n = 5$ are provided in Table 1.

It can be seen from Table 1 that for ***Rule# i*** power keeps increasing with the increase in δ^2, do not get smaller than α and ultimately converges to 1. But in case of ***Rules# ii*** and ***iii*** power increases with the increase in δ^2 for $\delta^2 \leq 2$ while in case of $\delta^2 > 2$ power starts decreasing with the increase in δ^2, eventually becomes smaller than the pre-specified value of α value and converges to 0 instead of 1 which is quite abnormal and unexpected. We term this issue as biasedness and non-monotonicity which are not desirable features for a control chart. A similar type of abnormality we have seen in the results of Jamali et al. (2007) for different sensitizing rules (cf. Tables 2-4 of Jamali et al. (2007)) but they have not addressed the solution to this issue. The results of our Table 1 and those of Jamali et al. (2007) clearly indicate that ***Rules# ii*** and ***iii*** and similar other rules may not be used in their independent capacities. However ***Rule# i*** does not face these types of issues as can be seen from Table 1.It means that ***Rule# i*** can be used independently with a control chart structure but ***Rules# ii*** and ***iii*** have no independent identity. However these problems in the form of unusual performance of different rules may be overcomed by combining the rules (e.g. ***Rules# ii*** and ***iii*** with ***i***) at the cost of increased false alarm rate as is observed by Does and schriever (1992) (cf. Table 2 of Does and schriever (1992)). As mentioned earlier we can manage this issue of inflated false alarm rates by appropriately adjusting the control limits for each rule but simultaneous use of more rules at a time may complicate the application and also makes the control chart structure unattractive for practitioners. Ideally each rule should have its own independent characteristics and should be capable to work with a control chart structure in its true spirit.

Table 1: Power of *Rules# i, ii* and *iii* for S^2 Chart (Upper Sided) at $n = 5$

Rules / δ^2	*Rule i*: $1/1$ $h_0 = 4.0628$	*Rule ii* : $2/3$ $h_{00} = 2.6688$	*Rule iii* : $4/5$ $h_{000} = 1.64755$
1	0.002700	0.002700	0.002700
1.05	0.005271	0.006088	0.006501
1.1	0.009352	0.012096	0.013495
1.15	0.015332	0.021637	0.024809
1.2	0.023535	0.035451	0.041254
1.25	0.034192	0.053949	0.063118
1.3	0.047416	0.077128	0.090080
1.4	0.081468	0.135464	0.155421
1.5	0.124575	0.203437	0.226720
2	0.397574	0.436760	0.409600
4	0.907406	0.122351	0.085876
10	0.996872	0.004113	0.002650
30	0.999959	0.000053	0.000033

2. Giving the Sensitizing Rules an Independent Identity

We may enhance detection ability of the design structure of a control chart by supporting them with some extra sensitizing rules and run rules schemes.Many authors have proposed different run rules/scheme for this purpose for example see Roberts (1958), Electric Handbook (1965), Bissell (1978), Wheeler (1983), Champ and Woodall (1987), Alwan(2000), Klien(2000), Khoo(2004), Montgomery (2005), Kourtas et al. (2007), Antzoulakos and Rakitzis (2008) and the references therein. Although implementation of such rules/scheme boost the sensitivity of control chart for smaller shifts as well but at the same time complicates their design structures and also creates some statistical problems.

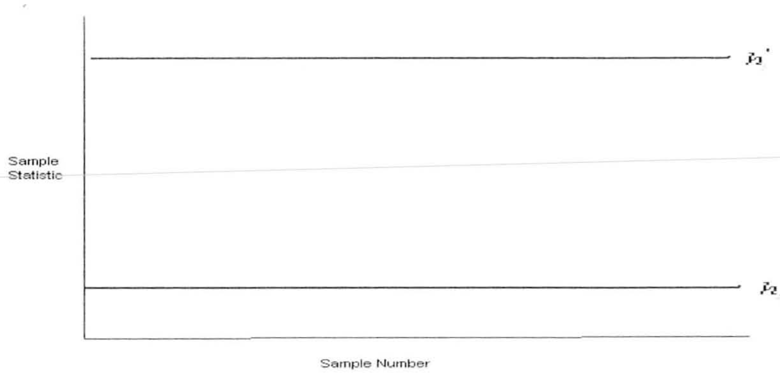
In order to overcome the problems with ***Rules# ii*** and ***iii*** as mentioned in Section 1 and to make these rules workable separately (without connecting their use with any other rule) we have redefined these rules and added some more similar rules to be used with control chart structures to enhance their performance. Table 2 consists of a set of different sensitizing rules which may be used with a control chart in their own autonomous capacities.

Table 2: Different Sensitizing Rules

Rule#	Notation	Rule Description
1	1/1	An out-of-control signal is received if one point falls out side the control (signaling) lines (h_1, h_1') defined either on one tail or both the tails of the sampling distribution of control charting statistic.
2	2/2	An out-of-control signal is received if two consecutive points fall outside the control (signaling) lines (h_2, h_2') defined either on one tail or both the tails of the sampling distribution of control charting statistic.
3	1/2	An out-of-control signal is received if at least one out of two consecutive points fall outside the control line (h_3, h_3') defined either on one tail or both the tails of the sampling distribution of control charting statistic.
4	3/3	An out-of-control signal is received if three consecutive points fall outside the control (signaling) lines (h_4, h_4') defined either on one tail or both the tails of the sampling distribution of control charting statistic.
5	2/3	An out-of-control signal is received if at least two out of three consecutive points fall outside the control (signaling)

		lines (h_5, h_5') defined either on one tail or both the tails of the sampling distribution of control charting statistic.
6	1/3	An out-of-control signal is received if at least one out of three consecutive points fall outside the control (signaling) lines (h_6, h_6') defined either on one tail or both the tails of the sampling distribution of control charting statistic
7	4/4	An out-of-control signal is received if four consecutive points fall outside the control (signaling) lines (h_7, h_7') defined either on one tail or both the tails of the sampling distribution of control charting statistic
8	3/4	An out-of-control signal is received if at least three out of four consecutive points fall outside the control (signaling) lines (h_8, h_8') defined either on one tail or both the tails of the sampling distribution of control charting statistic
9	2/4	An out-of-control signal is received if at least two out of four consecutive points fall outside the control (signaling) lines (h_9, h_9') defined either on one tail or both the tails of the sampling distribution of control charting statistic
10	5/5	An out-of-control signal is received if five consecutive points fall outside the control (signaling) lines (h_{10}, h_{10}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
11	4/5	An out-of-control signal is received if at least four out of five consecutive points fall outside the control (signaling) lines (h_{11}, h_{11}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
12	3/5	An out-of-control signal is received if at least three out of five consecutive points fall outside the control (signaling) lines (h_{12}, h_{12}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
13	6/6	An out-of-control signal is received if six consecutive points fall outside the control (signaling) lines (h_{13}, h_{13}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
14	5/6	An out-of-control signal is received if at least five out of six consecutive points fall outside the control (signaling) lines (h_{14}, h_{14}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
15	4/6	An out-of-control signal is received if at least four out of six consecutive points fall outside the control (signaling) lines (h_{15}, h_{15}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
16	7/7	An out-of-control signal is received if seven consecutive points fall outside the control (signaling) lines (h_{16}, h_{16}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
17	6/7	An out-of-control signal is received if at least six out of seven consecutive points fall outside the control (signaling) lines (h_{17}, h_{17}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
18	5/7	An out-of-control signal is received if at least five out of seven consecutive points fall outside the control (signaling) lines (h_{18}, h_{18}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
19	8/8	An out-of-control signal is received if eight consecutive points fall outside the control (signaling) lines (h_{19}, h_{19}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
20	7/8	An out-of-control signal is received if at least seven out of eight consecutive points fall outside the control (signaling) lines (h_{20}, h_{20}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
21	6/8	An out-of-control signal is received if at least six out of eight consecutive points fall outside the control (signaling) lines (h_{21}, h_{21}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
22	9/9	An out-of-control signal is received if nine consecutive points fall outside the control (signaling) lines (h_{22}, h_{22}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
23	8/9	An out-of-control signal is received if at least eight out of nine consecutive points fall outside the control (signaling) lines (h_{23}, h_{23}') defined either on one tail or both the tails of the sampling distribution of control charting statistic
24	7/9	An out-of-control signal is received if at least seven out of nine consecutive points fall outside the control (signaling) lines (h_{24}, h_{24}') defined either on one tail or both the tails of the sampling distribution of control charting statistic

The control lines $(h_i, h_i^{'})$ for $i=1,2,3,...,24$ used in Table 2 can be determined from the sampling distribution of the control charting statistic(s), depending upon the pre-specified value of α. Graphically it may be shown in the form of following figure in general:



The lines h_i and $h_i^{'}$ for $i=1\ 2,3,...,24$ may be define by using complete α on right tail of the sampling distribution for only upper sided control limit (i.e. $h_i^{'}$), on left tail of the sampling distribution for only lower sided control limit (i.e. h_i) and half of the α on both the tails for two sided control limits on a chart (i.e. $(h_i, h_i^{'})$).

Its to be noted that the phrase "at least" in the rules of Table 2 has a significance which gives an independent identification to each rules (which was in fact missing in ***Rules# i, ii*** and ***iii*** defined in Section 1. The inclusion on this phrase is in fact the suggested redefining (modification) for these run rules scheme which help overcoming the problems mentioned above in Section 1. It gives an attractive independence to each rule. The signaling limits h_i and $h_i^{'}$ are set according to the pre-specified value of α for and the choice of run rule schemes of Table 2. This is done using the mathematical expression $\alpha = \sum_{r<=m}^{m} \frac{m!}{r!(m-r)!} p^r (1-p)^{(m-r)}$, 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. This equation may be solved for p using any choice of α, r and m. The resulting value of p may help in picking up the appropriate quantile point on the sampling distribution of a control charting statistics for different r/m run rules schemes and hence computing respective control limits followed by evaluating power of detecting out-of-control signals.

Particularly we will consider the most commonly used Shewhart's control charts namely S^2, $\bar{X}$, R and S charts in this study but the application of these schemes may be generalized for any type of charts.

3. Performance Evaluation and Comparisons

In this section we evaluate the performance of different r/m run rules schemes using power as the performance criterion. It is a quite popular among practitioners to prefer statistical technique (e.g. a control chart) which has higher power and they use it in designing and finalizing their research proposals (cf. Mahoney and Magel (1996)). The literature supporting power evaluation criterion may be also be seen in Motgomery (2005), Albers and Kallenberg (2006), Riaz (2008).

Now we evaluate the performance of all the schemes of Table 2 by computing their powers and see whether these schemes are able to overcome the problems indicated in Section 1 and behave according to the expectations attached with them in Section 2. Based on this power evaluation we also compare the performance of each scheme with the other schemes and observe their superiority ranking with respect to each other. We have evaluated the performance of these S^2, $\bar{X}$, R and S charts in terms of power using all the schemes of Table 2 following the guidelines of Section 2 by assuming normality of the quality characteristic of interest say X (i.e. $X \sim N(\mu, \sigma)$).

For the said purposes we consider the upper sided control limits of S^2, $\bar{X}$, R and S charts for the sake of brevity and provide their power curves at α =0.0027 in Figures 1-4 (for ease in comparison) in which power is plotted verses different amounts of shifts in the process parameter. These shifts are considered in terms of $\delta\sigma$ for all the charts except S^2 where shifts are consider in terms of $\delta^2\sigma^2$. It is to be mentioned that for power computation we have developed an efficient code in $\mathbb{R}$ language and used it in our study.

We categorize the curves in three types referring to different rules as: some with $m-r=2$ (i.e. 1/3, 2/4, 3/5, 4/6, 5/7, 6/8, 7/9); some with i.e. 1/2, 2/3, 3/4, 4/5, 5/6, 6/7, 7/8, 8/9) and some with $m-r=0$ i.e. 1/1, 2/2, 3/3, 4/4, 5/5, 6/6, 7/7, 8/8, 9/9). It would provide an ease in display and discussion as well. As showing all 24 curves on a graph would complicate the display so we choose few representative curves using the *m-r* categorization of these rules. In the following Figures 1-4 we have chosen only six rules for display and the rest would be covered in the discussion.

Figure 1: Power Curve of Different Rules at n=5 for $\bar{X}$ Chart (upper sided)

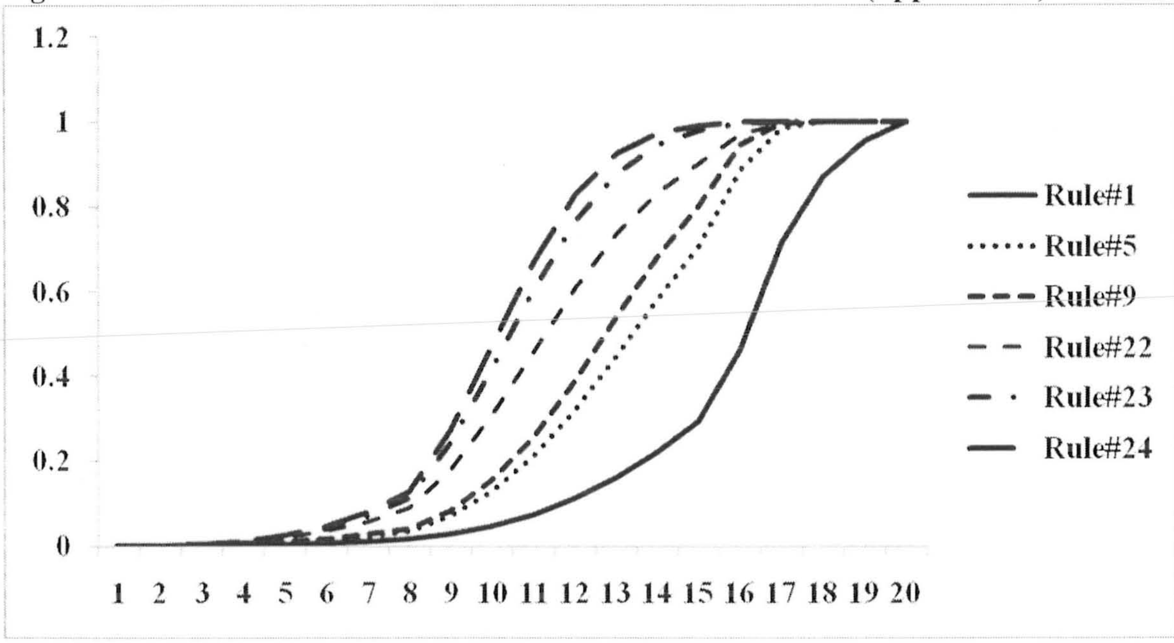
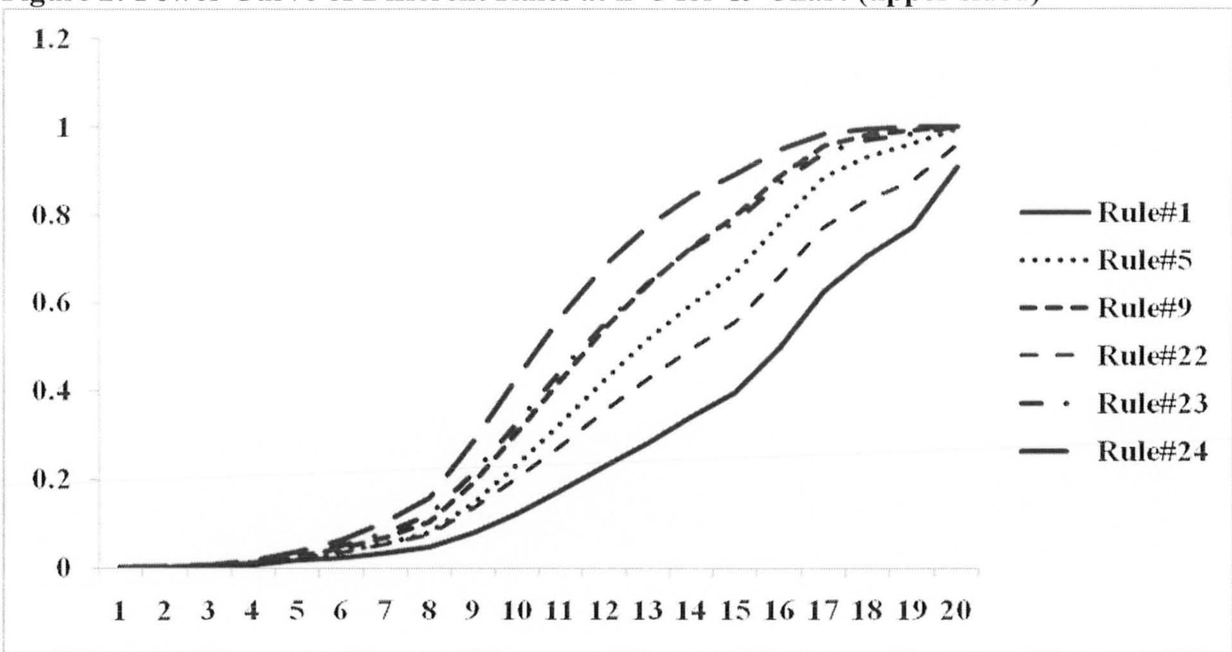


Figure 2: Power Curve of Different Rules at n=5 for R Chart (upper sided)



:

Figure 3: Power Curve of Different Rules at n=5 for S Chart (upper sided)

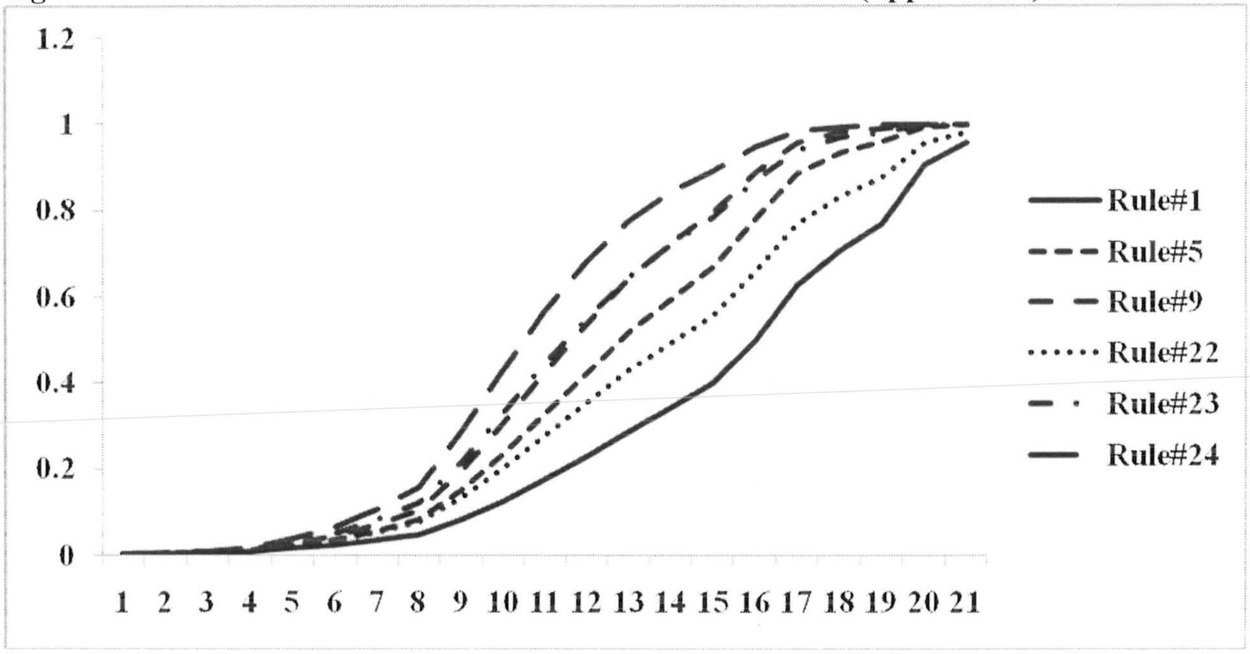
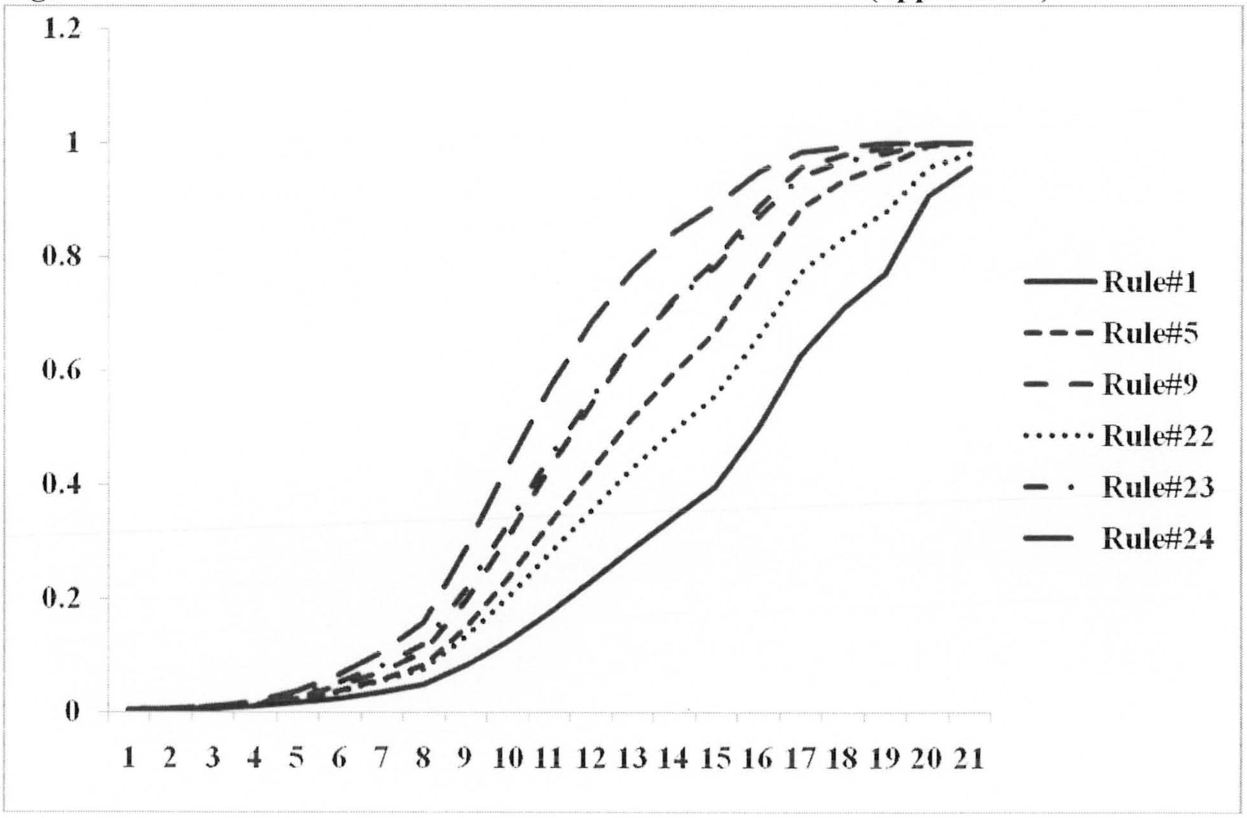


Figure 4: Power Curve of Different Rules at n=5 for S^2 Chart (upper sided)



For the construction of power curves the values of p used at α=0.0027 are $0.0027, 0.051962, 0.00135091, 0.1392477, 0.0303077, 0.000900811, 0.0079507, 0.0897928, 0.0215217, 0.3063887, 0.157665, 0.066885, 0.373159, 0.223105, 0.121913, 0.4295878, 0.282565, 0.177659, 0.4774418, 0.355424, 0.230386, 0.5183176, 0.382091, 0.27882$ for rules 1-24 respectively. The respective h_is used in the power curves of the four charts given above in Figures 1-4 are respectively provided in Tables 3-6 and these are given as:

Table 3: Control Lines of Different Sensitizing Rules at $n = 5$ for $\overline{X}$ Chart (Upper Sided)

Rule #	1	2	3	4	5	6	7	8
h_i	1.244216	0.727225	1.34154	0.484648	0.839103	1.395809	0.333448	0.600203
Rule #	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
h_i	0.904842	0.226341	0.449046	0.670552	0.144673	0.340664	0.521217	0.079346
Rule #	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
h_i	0.257255	0.41337	0.025301	0.190059	0.329854	-0.02054	0.134161	0.262224

Table 4: Control Lines of Different Sensitizing Rules at = 5 for S^2 Chart (Upper Sided)

Rule #	1	2	3	4	5	6	7	8
h_i	4.062793	2.348611	4.449728	1.734285	2.671919	4.674558	3.450777	2.012298
Rule	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
h_i	2.873965	1.204726	1.653841	2.194586	1.062636	1.423775	1.819466	0.957347
Rule	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
h_i	1.261611	1.575657	0.875718	1.09825	1.402035	0.810276	1.045198	1.270917

Table 5: Control Lines of Different Sensitizing Rules at = 5 for R Chart (Upper Sided)

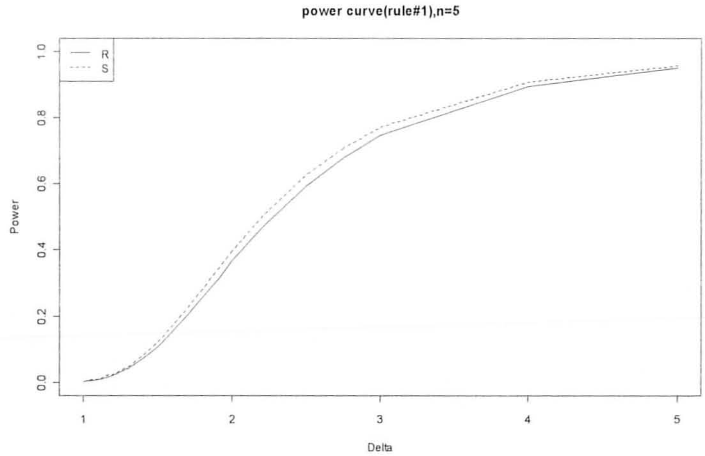
Rule #	1	2	3	4	5	6	7	8
h_i	5.1201	3.839592	5.376479	3.277863	4.106906	5.518022	2.943846	3.541697
Rule #	9	10	11	12	13	14	15	16
h_i	4.266604	2.716187	3.19796	3.706131	2.545364	2.959533	3.360344	2.413317
Rule #	17	18	19	20	21	22	23	24
h_i	2.781399	3.118335	2.306364	2.640111	2.936175	2.216399	2.523959	2.792012

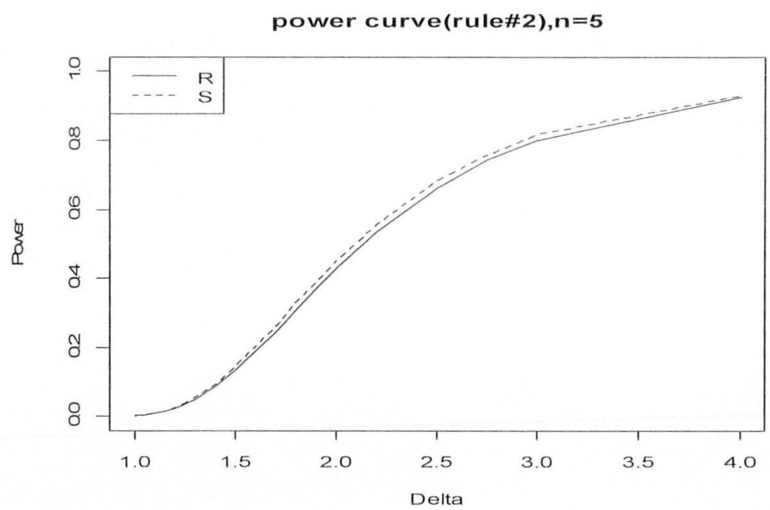
Table 6: Control Lines of Different Sensitizing Rules at $n = 5$ for S Chart (Upper Sided)

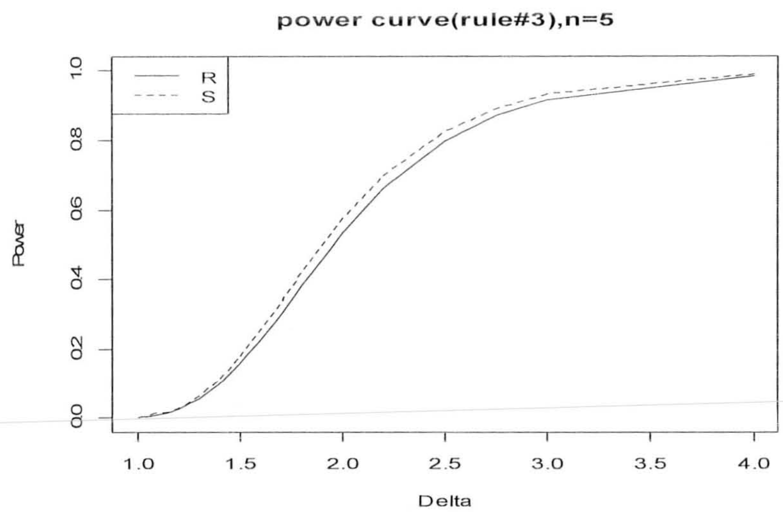
Rule #	1	2	3	4	5	6	7	8
h_i	2.015639	1.532514	2.109351	1.316924	1.634601	2.162075	1.187114	1.41858
Rule #	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
h_i	1.695258	1.097602	1.286017	1.481415	1.030844	1.193218	1.348877	0.978443
Rule #	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
h_i	1.123214	1.255253	0.935804	1.067778	1.184078	0.900156	1.02235	1.127351

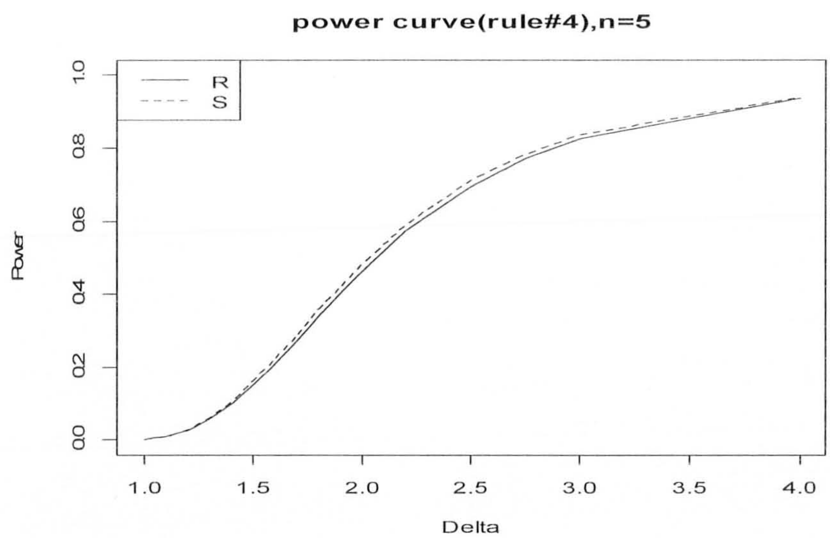
A similar behavior is observes for the other values of n and some more choices of α. The power curve analysis of all the 24 rules advocated us that all the schemes mentioned/described in Table 2 are unbiased and monotonic. Moreover the schemes where $m-r=2$ generally perform better followed by those where $m-r=1$ and the least efficient are those where $m-r=0$. Also within those schemes where $m-r=2$

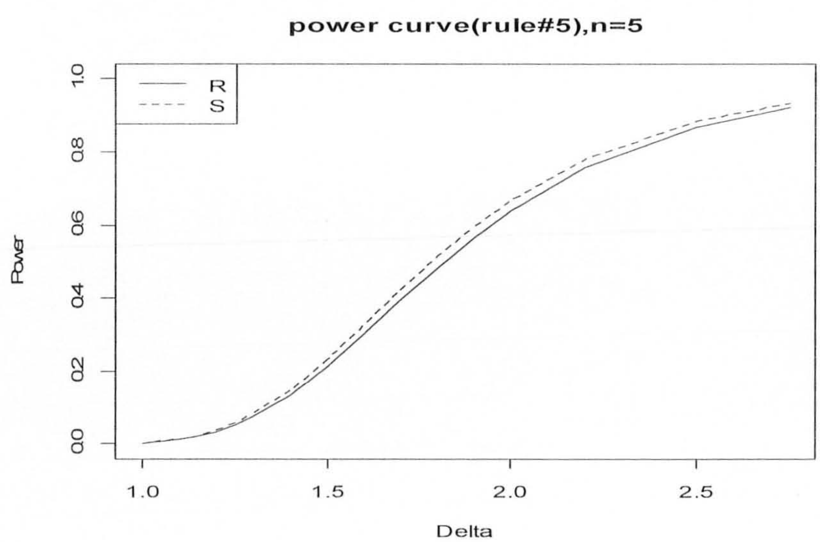
those consuming more consecutive points have in general better performance and the same comment is valid for those schemes where $m-r$ is 1 and 0. Furthermore we have observed that the S chart generally better performs better as compared the R chart for different r/m run rules scheme. This can be seen from the following few power curve graphs made for a comparison between R and S chart using different schemes at α =0.0027 and n=5.

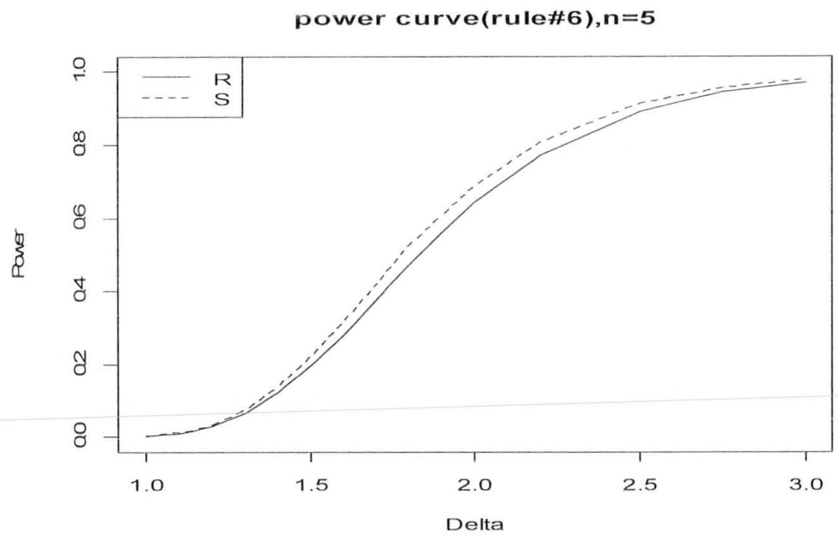


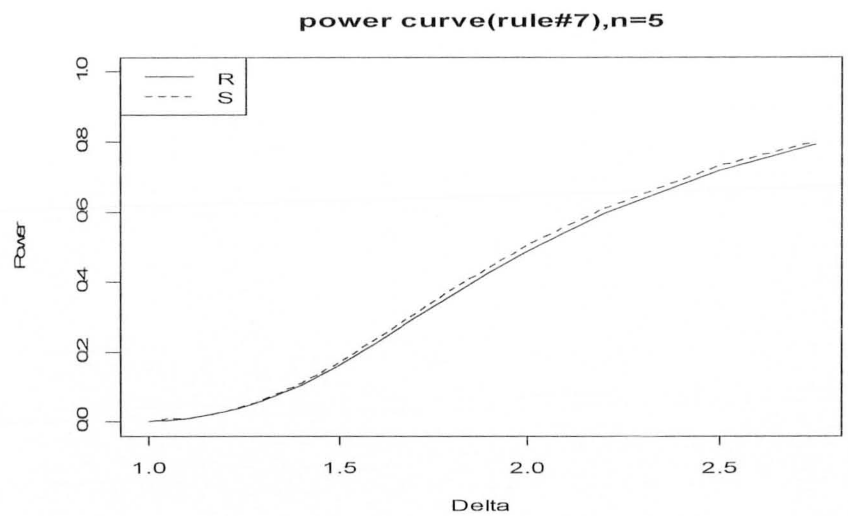


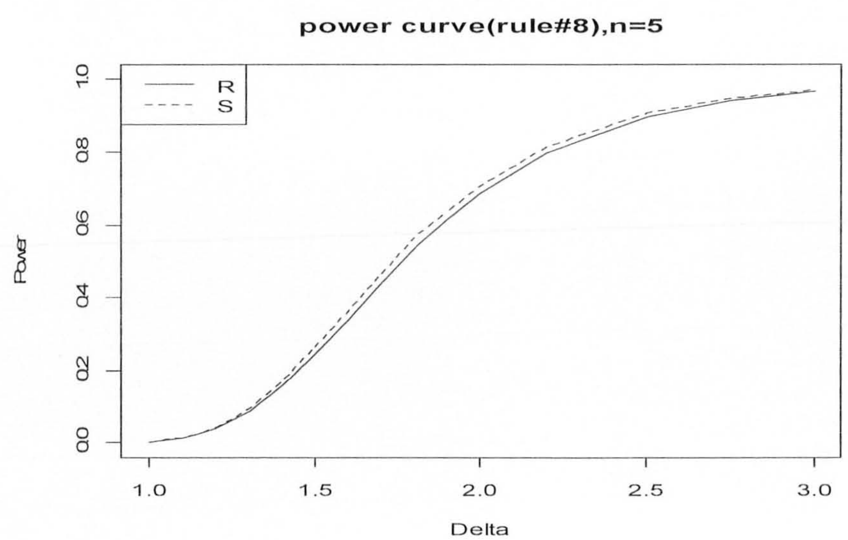

power curve(rule#3),n=5
R
S
Power
Delta
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0

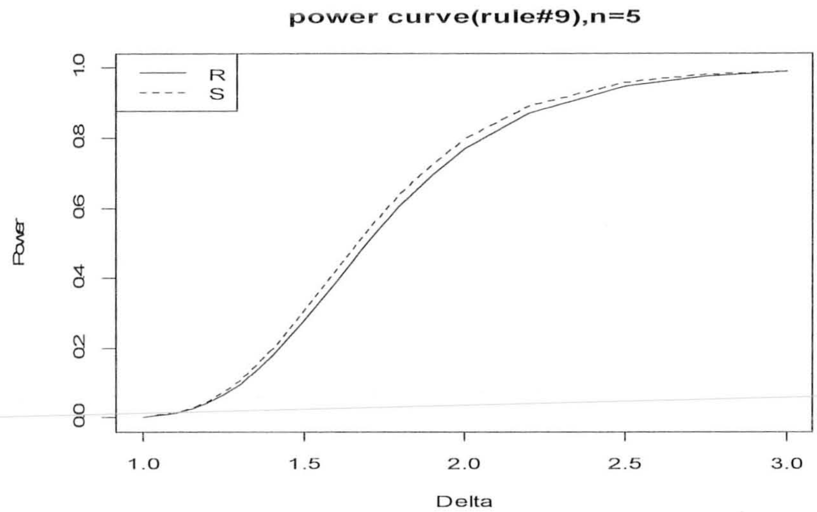

power curve(rule#4),n=5
R
S
Power
Delta
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0

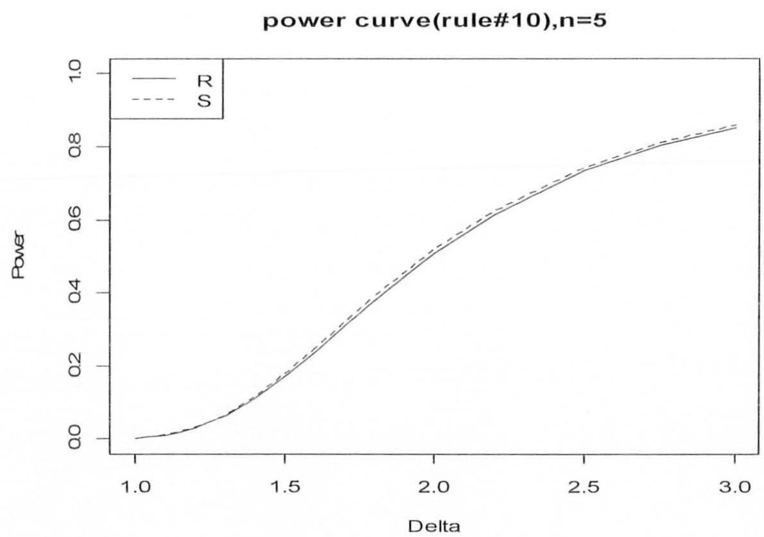

power curve(rule#5),n=5
R
S
Power
Delta
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5

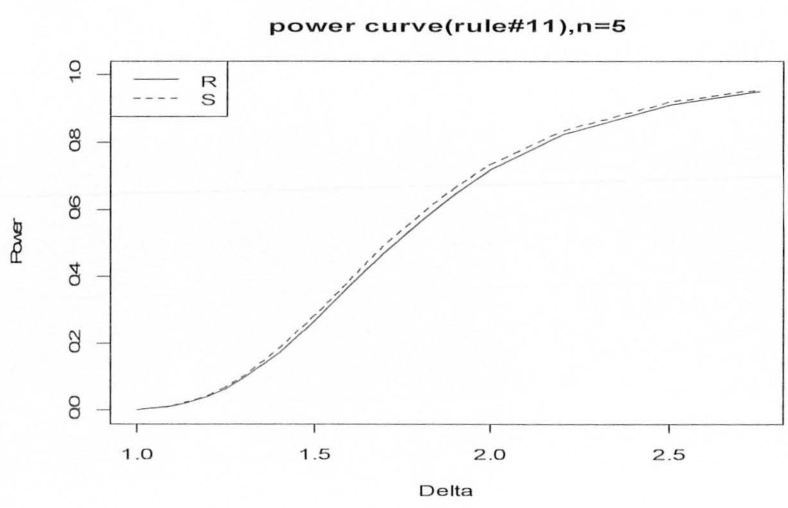

power curve(rule#6),n=5
R
S
Power
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
Delta

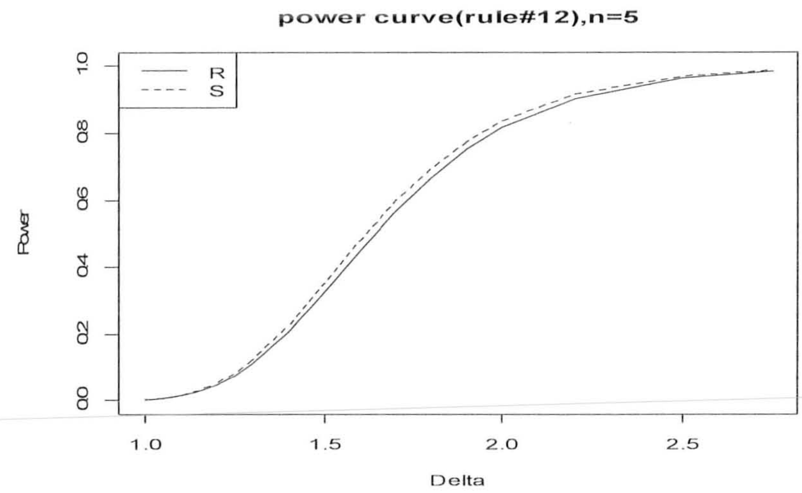

power curve(rule#7),n=5
R
S
Power
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
Delta


power curve(rule#8),n=5
R
S
Power
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
Delta


power curve(rule#9),n=5
R
S
Power
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
Delta


power curve(rule#10),n=5
R
S
Power
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
Delta


power curve(rule#11),n=5
R
S
Power
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
Delta

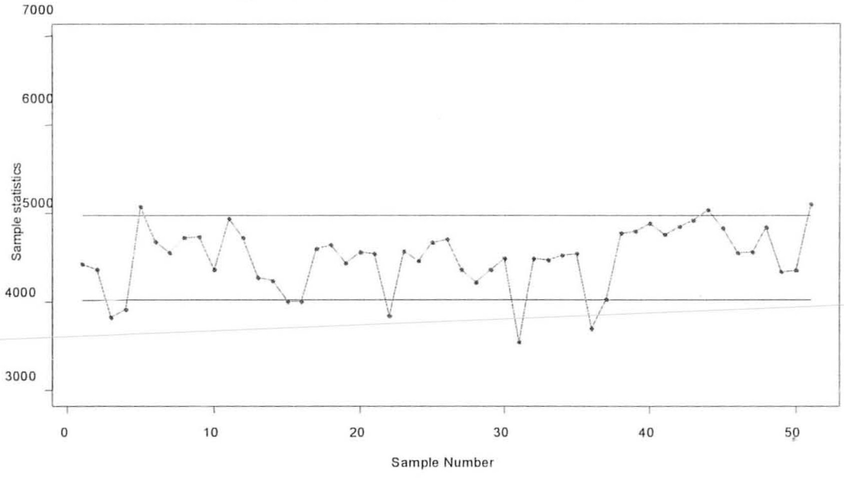


The same behavior we have observed for the other rules as well. To sum up we may infer that these suggested modifications for the runs rules schemes given in Table 2 have the ability to perform well in general in terms of their power efficiency.

4. Illustrative Example and Application

Besides evidences in terms of statistical efficiency it is always a good approach to test a technique on some real data for its practical implications. While working with practical datasets in industry, practitioners look for a direct application of these rules so that out-of-control signals may be received timely. For this purpose we consider here a dataset taken from Alwan (2000) which refers back to W.A. Shewhart's (1931) containing the data on 204 consecutive measurements of the electrical resistance of insulation in megohms. This data set may be seen on page 380 of Alwan (2000). To illustrate the application of the schemes given in Table 2 we apply two of them on the same data set to see what output they show. The other rules may also be applied very easily and for this purpose we have written an application code in $\mathbb{R}$ language for practitioner's convenience to use these rules on the real datasets (the code used for this purpose is available with authors and may be provided on request). The final control chart display along with full summary using our code for 1/1 and 1/2 schemes are given below in Figures 5 and 6. It is evident from these figures that the same out-of-control signals (10 in total) are received for 1/1 rule (cf. Figure 5) as those given by Alwan (2000) at α =0.0027. The application of 1/2 rule has given more (13 in total) out-of-control signals (cf. Figure 6) as compared to those of 1/1 keeping the false alarm rate fixed α at 0.0027. Similarly practitioners may easily apply the other rules on the real datasets.

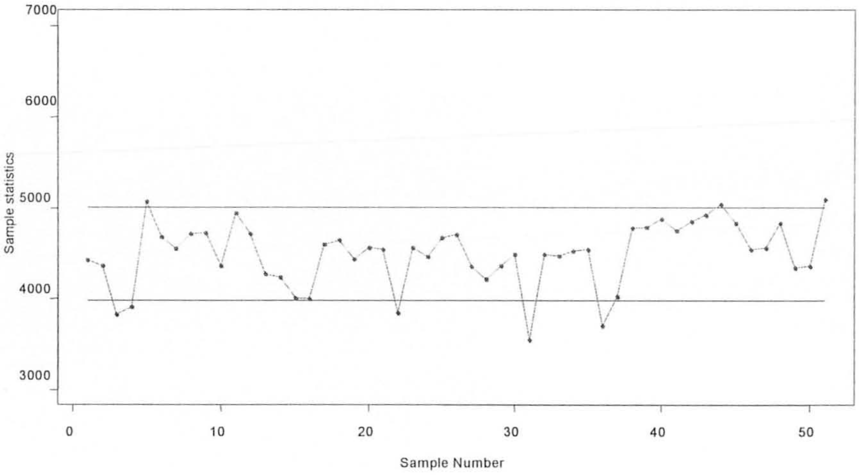
Figure 5: $\bar{X}$ Control Chart for megohm Data using 1/1 Rule



SUMMARY FOR OUT-OF-CONTROL SIGNALS (for Figure 5)

control chart : *Shewart Xbar*
control limits (T) : *4977.989 4018.364*
False Alarm Rate : *0.0027*
sample size : *4*
subgroup size : *51*
Out of control signals received at subgroups# : 3 4 5 15 16 22 31 36 44 51
Total# of out of control signals using 1 / 1 : 10

Figure 6: $\bar{X}$ Control Chart for megohm Data using 1/2 Rule



SUMMARY FOR OUT-OF-CONTROL SIGNALS (for Figure 6)

control chart : Shewart Xbar
control limits (T) : 5010.461 3985.892
False Alarm Rate : 0.0027
sample size : 4
subgroup size : 51
Out of control signals received at subgroups# : 3 4 5 6 22 23 31 32 36 37 44 45 51
Total# of out of control signals using 1 / 2 : 13

This application of the runs rules schemes exhibit that their implementation is quite efficient and also simple for the practitioner's use on the real data sets.

5. Summary, Conclusion, Recommendations and Future Proposals

Control charting is quite popular in SPC to identify problematic subgroups over a given sequence of time points. A timely signaling is a natural desire by the practitioners and for this purpose we have different types of control charts. Shewhart's type charts are efficient at detecting larger shifts and extra sensitizing runs rules schemes provide a good support to their design structures but at the same time they introduce some problems with their properties. These problems include: i)biasedness and non-monotonicity in case of separate use of each rule in general and hence no independent identity of any rule for different types of shifts; ii) need of simultaneous application of more rules at a time which results into an inflated false alarm rate and unattractive structures for practitioners use. By appropriately redefining these rules/schemes we have shown that each rule may have its own independent identification and hence may be used in its own capacity separately giving simplicity to the design structures of control charts and also overcoming the issues of inflated false alarm rate, biasedness and non-monotonicity.

The redefining of the runs rules schemes covered in this study may be extended for more rules as well. The application of these rules may also be implemented on EWMA and CUSUM charts. Also we may accomplish attribute control charts with these benefits of different runs rules schemes. The scope of this study may also be extended to the use of ranked set sampling as well as covering the Bayesian scenario and non-parametric setup (few of our future projects).

References

Albers, W. and W. C. M. Kallenberg (2006). Self-adapting control charts. Statistica Neerlandica, 60, 292-308.

Alwan, L. C. (2000). Statistical Process Analysis. McGraw-Hill International Editions: Singapore.

Antzoulakos, D. L. and Rakitzi, A. C. (2008) The modified r out of m control chart. Communication in Statistics-Simulations and Computations, 37, 396-408.

Biseell, A. F. (1978) . An attempt to unify the theory of quality control procedure. Bulletin in applied statistics, 5, 113-128

Champ, C. W. and Woodall, W. H. (1987). Exact results for Shewhart control charts with supplementary run rules. Tecnometrics, 29, 393-399.

Does, R. J. M. M. and Schriever, B.F. (1992). Variables control chart limits and tests for special causes. Statistica Neerlandica, 46, 229-245.

Jamali, A. S., Jinlin, L. and Chandio, A. F. (2007). Average Run Length Performance of Shewhart control chart with interpretation rules using markov chain approach.IEEE international conference on control and automation Guangzhou , China, May 13-June 1, 2007.

Khoo, M. B. C. (2004). Design of runs rules schemes. Quality Engineering, **16**, 27-4

Klein, M. (2000). Two Alternatives to the Shewhart X control chart. Journal of Quality Technology; 32, 427-431.

Koutras, M. V., Bersimis, S. and Maravelakis, P. E. (2007). Statistical process control using Shewhart control chart with supplementary runs rules. Methodology and Computing in Applied Probability, 9, 207-224.

Mahoney, M. and Magel, R. (1996). "Estimation of the Power of the Kruskal-Wallis Test". Biomtrical Journal, 38, 613-630.

Montgomery, D. C. (2005). Introduction o Statisyical Quality Control. John Wiley and Sons, New York (5th Edition).

Riaz, M. (2008). Improved and Robust Monitoring in Statistical Process Control. PhD thesis, University of Amsterdam.

Roberts, S. W. (1958). Properties of control charts zone test. The bell system technical journal, 37, 83-114.

Shewhart, W. (1931). Economic control of quality manufactured product, D. Van Nostrand, New York; reprinted by the American Society for Quality Control in 1980, Milwauker, WI.

Western Electric Company (1965). Statistical quality handbook, Indianapolis, Western Electric Corporation.

Wheeler , D. J. (1983) . Detecting a shift in process average. Tables of the power function for $\overline{X}$ chart. Journal of quality technology, 155-170.

Chapter 3

An Efficient Power Computation of r/m Runs Rules Schemes

In this chapter we shall develop a power computation code in $\mathbb{R}$ language which will provide an ease to researchers in designing their projects and doing their further studies using different existing and newly introduced sensitizing rules and runs rules schemes designed for varying Shewhart's control charting structures. This code will provide help to researcher to compute the power for different options of r/m rules/schemes. The said code will be flexible to be applied for any sample size, any false alarm rate, any type of control limits(one or two sided) and any amount of shift in process parameters for the four most commonly used Shewhart's type control charts namely S, R, $\overline{X}$, S^2 charts in case of both the specified and unspecified parameters cases. These mentioned benefits of our developed functional code are partially found in the features of the existing soft wares/packages and these may be enhanced by adding the features of our developed code as a function in their libraries dealing with quality control charting.

1. Introduction

There is no process which can avoid variations in its output. These variations are mainly of two types namely natural and unnatural. There is no solution to the natural variations (in-control process) and one is bound to live with them. For un-natural variations (out-of-control process) there always exists some special reasons which need to be identified and a timely identification of these special causes of variations help boosting the quality of a process. In order to address and differentiate these two types of violations we have a very useful instrument called Statistical Process Control (SPC) which contains very powerful tool like pareto chart, check sheets, cause and effect diagram, control chart etc. In this SPC tool kit control chart is the most important and commonly used tool due to its statistical framework. Shewart's type control charts are very popular because of this simplicity and these are good at detecting large shifts in process parameters. We may enhance their detection ability by supporting their design structures with some extra sensitizing rules and run rules schemes.

Many authors have proposed different run rules/scheme for this purpose for example see Alwan(2000), Klien(2000), Khoo(2004), Kourtas et al. (2007), Antzoulakos and Rakitzis (2008) and the references therein. Although implementation of such rules/scheme boost the sensitivity of control chart for smaller shifts as well but at the same time complicates their design structures. As a result performance evaluations of these types of chart become a difficult task. A very popular performance evaluation criterion of different types of control charting structure is power which is the probability of declaring a process as out-of-control when it is actually out-of-control. The most popular Shewart's type control charts include $\bar{X}, R, S \ and \ S^2$ charts which generally work with only one sensitizing rules (i.e. 1 out of 1 (1/1)) and mainly sensitive for larger shifts. By applying/attaching extra rules/scheme with their basic design structures give a push to their detection ability for smaller shifts but at cost of some issues including: i) difficulty in power computation due to complicated design structure; ii) inflation in 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 softwares/packages for power evaluation of different r/m rules; v) non-flexibility for any choice of false alarm rate.

There are software/packages available which accommodate few of the aforementioned runs rules/scheme and even these suffer the problems highlighted in (i) - (v) above. Therefore some efficient code is needed which addresses and overcomes these complications. In this study we intend to slightly modify the existing rules/schemes to overcome the issue (ii), (iii) & (v), and then develop an efficient code in R language which addresses issues (i) and (iv). The choice of R language is made due to its friendly environment; easy access; interactive software environment for statistical computation and graphics; its de-facto standard behavior and implementation link with S programming language; its wide application among statistician and researcher of other fields.

We now define/redefine the run rules scheme as: "At least r out of m consecutive (r/m) points fall out side 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 will consider $m = 1, 2, 3, \ldots, 9$ and for each m we shall consider the choices $r = m-2, m-1, m$ where $1 \le r \le m$. This way we have 24 possible rules of decision(i.e. 1/1, 1/2, 2/2, 1/3, 2/3, 3/3, 2/4, 3/4, 4/4, 3/5, 4/5, 5/5, 4/6, 5/6, 6/6, 5/7, 6/7, 7/7, 6/8, 7/8, 8/8, 7/9, 8/9, 9/9). In general terminology we will refer these rules as r/m rules". In this set of rules the phrase "at least" is in fact the suggested modification for these run rules scheme which help overcoming the above mentioned problems (ii), (iii) & (v) and giving an attractive independence to each rule. The signaling limit $h_{r/m}$ is set according to the pre-specified value of false alarm rate for a given value of n and it is done using the mathematical expression $\alpha = \sum_{r<=m}^{m} \frac{m!}{r!(m-r)!} p^r (1-p)^{(m-r)}$, 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.

Different softwers/packages available like MINITAB, SPSS, MATLAB, MAPLE, $\mathbb{R}$, S etc. do not have built-in functions for power computations of different runs rules schemes on different types of control charts and also do not allow flexibility for fixing α at the desired level. An $\mathbb{R}$ package for statistical quality control namely "qcc" is given by Scrucca (2004) to monitor the process characteristics. It is an efficient package for control charting of the charts like S, R, $\bar{X}$, S^2 charts but it works with only one sensitizing rule (i.e. 1/1) and misses the application of other rules which are really needed to address the smaller shifts. The power computations accommodated by Scrucca (2004) (and similarly for many other packages) are limited to very few rules/schemes like 1/1 (or so) while the other rules/schemes, which may be more powerful at detecting different shifts, are missing in their environments. Therefore we should have some efficient code of power computation for all the rules in general. Practitioners generally prefer statistical technique (e.g. a control chart) which has higher 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 them for their future studies. The literature supporting power evaluation crireion may be also be seen in Motgomery (2005), Albers and Kallenberg (2006), Riaz (2008).

Keeping in view the above discussion we are now convinced to develop a function in $\mathbb{R}$ language in such a way that the researcher may be able to use for power evaluation. The said functional code would be accommodative for any sample size (n), any false alarm rate (α), any amount of shift in the parameters of interest, any type of control limits (one or two sided) on the design structures of $\bar{X}, R, S$ *and* S^2 charts. The $\mathbb{R}$ code is written by using some built-in and some user-defined (defined by ourselves) 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 said code is provided in the following section.

2. Power Computation Code

```
#Power Computation Code for Xbar, R,S and S2 Charts (Lower, Upper and Two sided limits) for r out of m
(r/m) rules with any fixed false alarm rate for any sample size and a given amount of shift

        # Function to compute probability of a single point for any r/m rule

onepoint=function(alpha,r,m)
{
b=0;
a=c()
if(m-r==0)
 {
return(alpha^(1/r))
 }
if(m-r==1)
 {
for(i in 1:100000)
  {
  b=b+0.00001
  a[i]=b
  }
return(a[(m*(a^(r))*(1-a)+(a^(m))-alpha)>0 & (m*(a^(r))*(1-a)+(a^(m))-alpha)<=0.0001])
 }
 if(m-r==2)
 {
for(i in 1:100000)
  {
  b=b+0.00001
  a[i]=b
  }
q=NULL
q=factorial(m)/(factorial(r)*factorial(m-r))
return(a[ (q*(a^(r))*((1-a)^(m-r))+m*(a^(r+1))*(1-a)+(a^(m))-alpha)>0 &

(q*(a^(r))*((1-a)^(m-r))+m*(a^(r+1))*(1-a)+(a^(m))-alpha)<=0.0001])
 }
}
# End of Function to compute probability of a single point for any r/m rule

        # Function to compute control limits for any r/m rule

controllimit=function(chart,r,m,side,n,simu,mean,sd,alpha)
{
ff=onepoint(alpha,r,m)
p=ff[1]
range=c()
if(chart=="xbar"&& side=="U")
{
ucl=mean+(qnorm(1-p)*sd)/sqrt(n)
```

```
return(c(ucl))
}
if(chart=="xbar"&& side=="L")
{
lcl=mean+(qnorm(p)*sd)/sqrt(n)
return(c(lcl))
}
if(chart=="xbar"&& side=="T")
{
ucl=mean+(qnorm(1-(p/2))*sd)/sqrt(n)
lcl=mean+(qnorm(p/2)*sd)/sqrt(n)
return(c(ucl,lcl))
}
if(chart=="S" && side=="U")
{
ucl=(sqrt(qchisq(1-p,n-1)/(n-1)))*sd
return (c(ucl))
}
if(chart=="S" && side=="L")
{
lcl=(sqrt(qchisq(p,n-1)/(n-1)))*sd
return (c(lcl))
}
if(chart=="S" && side=="T")
{
ucl=(sqrt(qchisq(1-(p/2),n-1)/(n-1)))*sd
lcl=(sqrt(qchisq(p/2,n-1)/(n-1)))*sd
return (c(ucl,lcl))
}
if(chart=="S2" && side=="U")
{
ucl=(qchisq(1-p,n-1)/(n-1))*sd^2
return (c(ucl))
}
if(chart=="S2" && side=="L")
{
lcl=(qchisq(p,n-1)/(n-1))*sd^2
return (c(lcl))
}
if(chart=="S2" && side=="T")
{
ucl=(qchisq(1-(p/2),n-1)/(n-1))*sd^2
lcl=(qchisq(p/2,n-1)/(n-1))*sd^2
return (c(ucl,lcl))
}
if(chart=="R")
{
for(i in 1:1000000)
{
a=rnorm(n,mean,sd)
range[i]=(max(a)-min(a))/sd
}
if(side=="U")
{
ucl=sd*quantile(range,1-p)
return(ucl)
}
if(side=="L")
{
lcl=sd*quantile(range,p)
return(lcl)
}
if(side=="T")
{
lcl=sd*quantile(range,1-(p/2))
ucl=sd*quantile(range,(p/2))
return(c(ucl,lcl))
}
}
}
```

End of Function to compute conmtrol limits for any r/m rule

 # Function to compute power for any r/m rule

```
power=function(chart,r,m,side,n,simu,mean,sd,alpha,delta)
{
h=controllimit(chart,r,m,side,n,simu,mean,sd,alpha)
c1=0
for(i in 1:simu)
{
sum1=0;
for(i in 1:m)
{
if(chart=="xbar")
{
gen=mean(rnorm(n,mean+delta,sd))
}
if(chart=="S")
{
gen=sqrt(var(rnorm(n,mean,sd*delta)))
}
if(chart=="S2")
{
gen=(var(rnorm(n,mean,sd*sqrt(delta))))
}
if(chart=="R")
{
a=rnorm(n,mean,sd*delta)
gen=max(a)-min(a)
}
if(side=="T")
{
if(gen>h[1]||gen<h[2])
{
sum1=sum1+1;
}
}
if(side=="U")
{
if(gen>h)
{
sum1=sum1+1;
}
}
if(side=="L")
{
if(gen<h)
{
sum1=sum1+1;
}
}
}
if(sum1>=r)
{
c1=c1+1;
}
}
pro=round(c1/simu,4);

# Displaying the final output

cat("\n","\n","\t","SUMMARY FOR OUT-OF-CONTROL PROBABILITY","\n","\n","\t","Control Chart","\t",
":","\t",chart,"\n","\t","Control Limit", "(",side,")",
":",controllimit(chart,r,m,side,n,simu,mean,sd,alpha),"\n", "\t","Rule Type", "\t",
":","\t",r,"/",m,"\n","\t","False Alarm Rate :","\t",alpha,"\n","\t","Sample Size","\t", ":
","\t",n,"\n","\t","Delta","\t","\t"," : ","\t",delta,"\n","\t","Power","\t","\t", ": ","\t",pro,"\n")
}
# End of Function to compute power for any r/m rule
```

3. Description of the Code

In this section we provide a detailed description of our code in terms of its development and functionality for the desired purposes. This description will help researcher to implement it easily for their desired use.

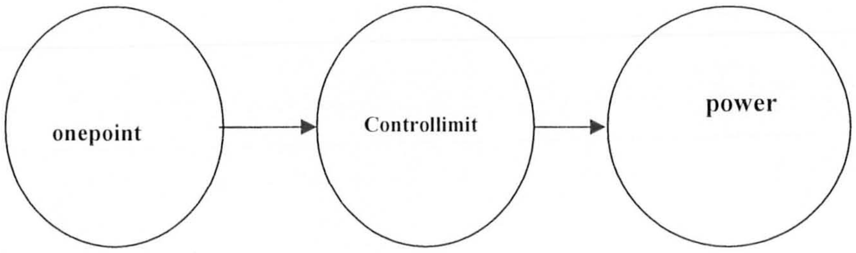
This code mainly consists of three user defined (our own developed) functions namely "onepoint", "controllimit" and "power" to meet the desired objectives. Now we describe all the three functions one by one.

The function **"onepoint"** (onepoint=function(alpha, r, m)) provides the probability of a single point falling outside the respective signaling limits. It returns the optimal solution for one point probability to fix α at the desired level by applying the index search approach using the arguments "alpha", "r"and "m". The argument "alpha" is pre-specified false alarm rate and it can 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)) returns 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 "delta". The argument "chart" can take any 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 any 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 arguments "sd" and "mean" represent the specified values of standard deviation and mean respectively and are used in the control limits construction. The other arguments are as defined earlier.

The function **"power"** (power=function(chart,r,m,side,n,simu,mean,sd,alpha,delta)) is the front end function for researcher. The arguments used in this function are the , "chart", "r", "m", "alpha", "side","n "simu" ,"mean", "sd","alpha", and "delta". 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 $\bar{X}$ 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. All the other arguments used in **"power"** function are as defined earlier and it calls the function "controllimit" to use the control limits. This function obtains the corresponding statistic values according to the choice of "chart" and then these statistics are consecutively compared with the values of control limit obtained form the function "controllimit". The number of times the 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). This summary output is shown in ℝ editor or console in an attractive format for the user.

To further clarify the flow of our code we provide here a flow diagram:



4. Execution Procedure of the Code

In this section we list down the necessary steps which will help the users/researchers to adopt our developed code for computing the control limits of a control chart at a given sample size and finally evaluating the detection power ability.

i) Run the code of Section 2 in $\mathbb{R}$ 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 $\mathbb{R}$ library as a function of quality control package).

ii) To get the final output in the form of 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

power(chart="?",r=?,m=?,side="?",n=?,simu=?,mean=?,sd=?,alpha=?,delta=?)

where each '?' refers to a specific choice for that particular argument e.g.

power(chart="xbar",r=1,m=1,side="T",n=5,simu=1000000,mean=0,sd=1,alpha=0.0027,delta=0)

is for $\bar{X}$ chart using 1/1 rule at α=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 Section 3 as per requirements.

iii) After the execution of **"power"** function we will get the final output of the data which researchers/users may want in the form of summary display in R-console providing the information of control chart type, control limits type (one or two sided), run rule type (r/m), false alarm rate, delta and power. This summary display will be easier and very attractive for researcher to quickly have an idea of the chart's ability.

5. 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 few of the runs rules schemes at different false alarm rates using some choices of delta (shifts) for the control charts considered in this study and see what output display is shown 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 1/1 rule on $\bar{X}$ chart we upload the code of Section 2 in $\mathbb{R}$ console and execute the following statement:

```
power(chart="xbar",r=1,m=1,side="T",n=5,simu=100000,mean=0,sd=1,alpha=0.0027,delta=0)
```

which gives the final output in $\mathbb{R}$ console as:

```
SUMMARY FOR OUT-OF-CONTROL PROBABILITY
    Control Chart  :     xbar
    Control Limit ( U ) : 304.9716
    Rule Type      :     1 / 1
    False Alarm Rate :     0.0027
    Sample Size    :     5
    Delta          :     0
    Power          :     0.0026
```

We can see that the code has given almost the same false alarm rate as desired. The accuracy level may be increased by increasing the number of simulations.
For some more cases of different control charts using varying runs rules/schemes for different amounts of shifts at a given false alarm rate we run our power code for some choices of sample sizes and see what results we finally have:

```
SUMMARY FOR OUT-OF-CONTROL PROBABILITY
    Control Chart  :     xbar
    Control Limit ( T ) : 1.341630 -1.341630
    Rule Type      :     1 / 1
    False Alarm Rate :     0.0027
    Sample Size    :     5
    Delta          :     1
    Power          :     0.2216
```

SUMMARY FOR OUT-OF-CONTROL PROBABILITY

Control Chart : xbar
Control Limit (T) : 1.341630 -1.341630
Rule Type : 1 / 1
False Alarm Rate : 0.0027
Sample Size : 5
Delta : 2
Power : 0.929

SUMMARY FOR OUT-OF-CONTROL PROBABILITY

Control Chart : S2
Control Limit (U) : 3.34365
Rule Type : 1 / 1
False Alarm Rate : 0.0027
Sample Size : 7
Delta : 1
Power : 0.0028

SUMMARY FOR OUT-OF-CONTROL PROBABILITY

Control Chart : S2
Control Limit (U) : 3.34365
Rule Type : 1 / 1
False Alarm Rate : 0.0027
Sample Size : 7
Delta : 2
Power : 0.123

SUMMARY FOR OUT-OF-CONTROL PROBABILITY

Control Chart : S
Control Limit (U) : 2.382778
Rule Type : 4 / 6
False Alarm Rate : 0.01
Sample Size : 10
Delta : 1.5
Power : 0.8644

Similarly any option may be entered in the **"power"** function to compute out-o-control probability (i.e. power) for any chart at any false alarm rate with a given sample size and any amount of shift.

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 S, R, $\overline{X}$, S^2 control charts with any run rule scheme under consideration at any false alarm rate and for any given amount of shift in the process parameters, which is not the case in general with the built-in function of other softwares/packages. It may also be noted that for those cases where theoretical (or otherwise) results for power of a control chart are available, our 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 S^2, $\overline{X}$, R and S charts given by Alwan(2000), Klien(2000) Khoo (2004), Motgomery (2005), Kourtas et al. (2007), Antzoulakos and Rakitzis (2008), Riaz (2008) etc.). To further enhance the accuracy of our results we may increase the number of simulations.

We may, therefore, sum up that our developed power computation code will be of great use for the practitioners and researchers who may make use of it in their proposals/projects where power of a statistical technique is used as performance criterion.

6. 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 scheme to compute power of a control chart. The existing packages/softwares generally work with few sensitizing rules/runs rules schemes and at very limited choices of false alarm rates. We have developed a functional code which can easily do it for many rules/schemes and for any choice of false alarm rate (this we have done for the four charts under discussion namely S, R, $\overline{X}$, S^2 charts at the moment and it may be done for other charts as well (one of our future projects)). Our code provides an attractive summary display of power along with the respective control limits of the corresponding control chart at any false alarm rate and for any amount of shift in $\mathbb{R}$ console directly.

This code may be easily modified for the distributions other than normal. Moreover it may be extended for the EWMA and CUSUM charts (one of our future projects). Also this code may be modified for Average Run Length (ARL) study of 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 (one of our future projects).

Keeping in view the above mentioned features of our proposed functional code we suggest its inclusion in the library of $\mathbb{R}$ language as a quality control tool/package with the name **"power"** just like **"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 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)).

References

Albers, W. and W. C. M. Kallenberg (2006). Self-adapting control charts. Statistica Neerlandica, 60, 292-308.

Alwan, L. C. (2000). Statistical Process Analysis. McGraw-Hill International Editions: Singapore.

Antzoulakos, D. L. and Rakitzi, A. C. (2008) The modified r out of m control chart. Communication in Statistics-Simulations and Computations, 37, 396-408.

Khoo, M. B. C. (2004). Design of runs rules schemes. Quality Engineering, 1**6**, 27-4

Klein, M. (2000). Two Alternatives to the Shewhart X control chart. Journal of Quality Technology; 32, 427-431.

Koutras, M. V., Bersimis, S. and Maravelakis, P. E. (2007). Statistical process control using Shewhart control chart with supplementary runs rules. Methodology and Computing in Applied Probability, 9, 207-224.

Mahoney, M. and Magel, R. (1996). "Estimation of the Power of the Kruskal-Wallis Test". Biomtrical Journal, 38, 613-630.

Montgomery, D. C. (2005). Introduction o Statisyical Quality Control. John Wiley and Sons, New York (5th Edition).

Riaz, M. (2008). Improved and Robust Monitoring in Statistical Process Control. PhD thesis, University of Amsterdam.

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

Chapter 4

An Easy Implementation of r/m Runs Rules Schemes for Practitioners

In this chapter we shall develop a functional code in $\mathbb{R}$ language which will provide an ease to practitioners in applying different existing and newly introduced sensitizing rules and runs rules schemes designed for varying Shewhart's control charting structures. This code will help identifying out-of-control signals very easily using different r/m rules. The said code will be flexible to be applied for any number of subgroups, any sample size, any false alarm rate and any type of control limits(one or two sided) for the four most commonly used .control charts namely S, R, $\overline{X}$, S^2 charts in case of both the specified and unspecified parameters cases. These mentioned benefits of our developed functional code are partially found in the features of the existing softwares/packages.

1. Introduction

It is generally desirable to have an easy implementation code of statistical/mathematical procedures for the ease of practitioners who are using them in the form of different softwares. If software does not support the application of certain technique(s) then the user of these techniques start avoiding them in their practice. Consequently these methods lose their popularity even if these are current and very efficient in other aspects of their relevant subject. For this reason practitioner like to have some softwares or functional codes which are easy to use and at the same time carry the efficiency of the particular technique in use with its true spirit.

Statistical quality control is an area of statistics where process variations are generally monitored through control charting procedures. Shewart's type control charts are generally based on normality assumption and mainly focus on larger shifts in the process parameters (cf. Motgomery (2005)) and the most popular of these charts include $\bar{X}, R, S, and\ S^2$ charts. Initially these charts were supported by very few sensitizing rules but later on a number of run rules/schemes were introduced to enhance their performance for smaller shifts as well. These rules are generally applied in combination with each other in order to keep them sensitive both for the larger and smaller shifts. A list of these sensitizing rules/run rules schemes may be seen in literature such as Alwan(2000), Klien(2000), Khoo(2004), Kourtas et al. (2007), Antzoulakos and Rakitzis (2008) and the references therein. A practitioner wants an easy implementation of these newly launched sensitizing rules/run rules scheme on the design structures of different charts like $\bar{X}, R, S\ and\ S^2$. For this purpose there are softwers/packages/languages available like MINITAB, SPSS, MATLAB, MAPLE, $\mathbb{R}$,S etc. These softwers are limited to the application of few of these runs rules/schemes.

In order to enhance the performance of different control charting structures (particularly for smaller shifts) the aforementioned sensitizing rules and runs rules schemes are very attractive in terms of their statistical properties for the said purpose but their implementation for practitioners is not very simple task as they complicate the control charting structure. Therefore for practical application of these rules/schemes in an easy mode for practitioners is always desired and this may be done with the help of softwares/packages or functional codes developed for the desired purposes. Although the softwraes/packages, as mentioned above, provide limited application of some of these rules/schemes but at the same time these suffer some other serious problems including: i) creating biasedness issue in case of an independent application of each rule separately in a given situation; ii) not very flexible for any amount of false alarm rate; iii) not speedily updating their features according to the development of newly designed sensitizing rules and runs rules schemes; iv) inflate the false alarm rate in case of simultaneous use of different rules; v) not giving an attractive and easy mode of application to the practitioners.

In this article we intend to slightly modify these rules in order to overcome the above mentioned issues and then develop their application code in $\mathbb{R}$ language for the ease of practitioners. The reason to choose $\mathbb{R}$ language here is that it is provides a user friendly programming environment; it is free of cost and very easy to access. Moreover it grants the software environment for statistical computing and graphics. It is the implementation of S programming language and it behaves de-facto standard among statistician for the development of statistical software. It is also widely used for statistical data analysis and packages for other fields like biological science, social science etc. are also developed in it.

We shall cover here 24 run rules scheme and an $\mathbb{R}$ code will be developed for their easy implementation. Following are the rules which will cover in this study:

- At least r out of m consecutive (r/m) points fall out side 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 will consider $m = 1, 2, 3, \ldots, 9$ and for each m we shall take $r = m-2, m-1, m$ where $1 \le r \le m$. This way we have 24 possible rules of decision(i.e. 1/1, 1/2, 2/2, 1/3, 2/3, 3/3, 2/4, 3/4, 4/4, 3/5, 4/5, 5/5, 4/6, 5/6, 6/6, 5/7, 6/7, 7/7, 6/8, 7/8, 8/8, 7/9, 8/9, 9/9). In general terminology we will refer these rules as r/m rules.

In the above mentioned set of rules the phrase "at least" is in fact the suggested modification for these run rules scheme which help overcoming the above mentioned problems and giving an attractive independence to each rule. The signaling limit $h_{r/m}$ is set according to the pre-specified value of false alarm rate for a given value of n and it is done using the following mathematical expression:

$$\alpha = \sum_{r<=m}^{m} \frac{m!}{r!(m-r)!} p^r (1-p)^{(m-r)} \qquad (1)$$

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.

An $\mathbb{R}$ package for statistical quality control namely "qcc" is given by Scrucca (2004) to monitor the process characteristics. It is an efficient package for control charting of the charts like S, R, $\bar{X}$, S^2 charts but it works with only one sensitizing rule (i.e. 1/1) and misses the application of other rules which are really needed to address smaller shifts. Murrell and Gardiner (2009) worked on graphicsQC package for $\mathbb{R}$ to provide functions which help producing and comparing the graphical display and giving the report of final results.

Keeping in view the aforementioned limitations of the existing sotwares/packages and taking the inspirations from Scrucca (2004) and Murrell and Gardiner (2009) we now develop a function in $\mathbb{R}$ language in such a way that the practitioner may be able to use it easily for any number of subgroups (k), any sample size (n), any false alarm rate (α), any type of control limits (one or two sided) on the design structures of $\bar{X}, R, S$ *and* S^2 charts for the cases of known and unknown standards (parameters). The $\mathbb{R}$ code is written by using some built-in and some user-defined (defined by ourselves) 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 said code is provided in the following section.

2. Actual Code for Practitioner's

```
# Practitioners Code for Xbar, R,S and S2 Charts(Lower, Upper and Two sided limits) for r out of m (r/m) rules with fixed false alarm rates for any sample size and any number of subgroups

b=0 #initialization of counter
a=c() #data vector declaration to store the possible values of one point probability at the desired accuracy level
ff=c()#data vector declaration to store single point probability returned by the function onepoint
range=c()#data vector declaration to generate empirical dist of Range
gen1=c()#data vector declaration to store the values of  sample mean statistics
gen2=c()#data vector declaration to store the values of  sample S statistics
gen3=c()#data vector declaration to store the values of  sample S2 statistics
gen4=c()#data vector declaration to store the values of  sample Range statistics
gen=c()#data vector declaration to store the values  of  different sample Range statistics
st=NULL

          # Function to compute probability of a single point for any r/m rule

onepoint=function(alpha,r,m)
 {
if(m-r==0)
 {
return(alpha^(1/r))
 }

if(m-r==1)
 {
for(i in 1:100000)
  {
  b=b+0.00001
  a[i]=b
  }
return(a[(m*(a^(r))*(1-a)+(a^(m))-alpha)>0 & (m*(a^(r))*(1-a)+(a^(m))-alpha)<=0.0001])
 }
 if(m-r==2)
 {

for(i in 1:100000)
  {
  b=b+0.00001
  a[i]=b
  }
 q=NULL
q=factorial(m)/(factorial(r)*factorial(m-r))
return(a[ ( m*a*(1-a)^(m-r)+q*a^(m-r+1)+a^(m))>0 & ( m*a*(1-a)^(m-r)+q*a^(m-r+1)+a^(m))<=0.001])
 }
 }
# end of Function to compute probability of a single point for any r/m rule
```

Function to compute control limits of different Control Charts

```
controllimt=function(data,chart,r,m,alpha,side,sd,mean,para)
    {
      n=length(data[1,])
      ff=onepoint(alpha,r,m)
      p=ff[1]
for(i in 1:length(data[,1]))
      {
      gen1[i]=rowMeans(data[i,])
      gen2[i]=sqrt(var(t(data[i,])))
      gen3[i]=(var(t(data[i,])))
      gen4[i]=max(t(data[i,]))-min(t(data[i,]))
      }

c4=c(0.7979,0.8862,0.9213,0.94,0.9515,0.9594,0.9650,0.9693,0.9727,0.9754,0.9776,0.9794,0.9810,0.9823,0.9835,0.9845,0.9854,0.9
862,0.9869,0.9876,0.9882,0.9887,0.9892,0.9896) #unbiasing constants of S chart for n=2-25
uc=c4[length(data[1,])-1]#selection of desired c4 according to sample size (n)
d2=c(1.128,1.693,2.059,2.326,2.534,2.704,2.847,2.970,3.078,3.173,3.258,3.336,3.407,3.472,3.532,3.588,3.640,3.689,3.735,3.778,3.8
19,3.858,3.895,3.931)  #unbiasing constants of R Chart for n=2-25
ud=d2[length(data[1,])-1]#selection of desired d2 according to sample size (n)
for(h in 1:10000)# Empirical Dist of R
  {
   a=rnorm(n,mean,sd)
   range[h]=(max(a)-min(a))/sd
  }

xbaru=qnorm(1-p)/sqrt(length(data[1,]))
xbarl=qnorm(p)/sqrt(length(data[1,]))
xbartu=qnorm(1-(p/2))/sqrt(length(data[1,]))
xbartl=qnorm(p/2)/sqrt(length(data[1,]))

su=sqrt(qchisq(1-p,n-1)/(n-1))
sl=sqrt(qchisq(p,n-1)/(n-1))
stu=sqrt(qchisq(1-(p/2),n-1)/(n-1))
stl=sqrt(qchisq(p/2,n-1)/(n-1))

s2u=(qchisq(1-p,n-1)/(n-1))
s2l=(qchisq(p,n-1)/(n-1))
s2tu=(qchisq(1-(p/2),n-1)/(n-1))
s2tl=(qchisq(p/2,n-1)/(n-1))

ru=quantile(range,1-p)
rl=quantile(range,p)
rtu=quantile(range,1-(p/2))
rtl=quantile(range,p/2)

if(chart=="Xbar" && side=="U" && para=="yes")
{
ucl=(sd*xbaru)+mean
return(ucl)
}
if(chart=="Xbar" && side=="U" && para=="no")
{
ucl=((mean(gen4)/ud)*xbaru)+mean(gen1)
return(ucl)
}
if(chart=="Xbar" && side=="L"  && para=="yes" )
{
lcl=mean+(sd*xbarl)
return(lcl)
}
if(chart=="Xbar" && side=="L"  && para=="no" )
{
lcl=(mean(gen4)/ud*xbaru)+mean(gen1)
return(lcl)
}
if(chart=="Xbar" && side=="T" && para=="yes")
{
ucl=(sd*xbartu)+mean
```

```
lcl=(sd*xbartl)+mean
return(c(ucl,lcl))
}
if(chart=="Xbar" && side=="T" && para=="no")
{
ucl=(mean(gen4)/ud*xbartu)+mean(gen1)
lcl=(mean(gen4)/ud*xbartl)+mean(gen1)
return(c(ucl,lcl))
}
if(chart=="S" && side=="U" && para=="yes")
{
ucl=sd*su
return(ucl)
}
if(chart=="S" && side=="U" && para=="no")
{
ucl=(su*mean(gen2))/uc
return(ucl)
}

if(chart=="S" && side=="L" && para=="yes")
{
sl=sqrt(qchisq(p,n-1)/(n-1))
lcl=sd*sl
return(lcl)
}
if(chart=="S" && side=="L" && para=="no")
{
lcl=(sl*mean(gen2))/uc
return(lcl)
}
if(chart=="S" && side=="T" && para=="yes")
{
ucl=sd*stu
lcl=sd*stl
return(c(ucl,lcl))
}
if(chart=="S" && side=="T" && para=="no")
{
ucl=mean(gen2)*stu
lcl=mean(gen2)*stl
return(c(ucl,lcl))
}
if(chart=="S2" && side=="U" && para=="yes")
{
ucl=s2u*sd^2
return(ucl)
}
if(chart=="S2" && side=="U" && para=="no")
{
ucl=s2u*mean(gen3)
return(ucl)
}
if(chart=="S2" && side=="L" && para=="yes")
{
lcl=s2l*sd^2
return(lcl)
}
if(chart=="S2" && side=="L" && para=="no")
{
lcl=s2l*mean(gen3)
return(lcl)
}
if(chart=="S2" && side=="T" && para=="yes")
{
ucl=s2tu*sd^2
lcl=s2tl*sd^2
return(c(ucl,lcl))
}
if(chart=="S2" && side=="T" && para=="no")
```

```
{
ucl=s2tu*mean(gen3)
lcl=s2tl*mean(gen3)
return(c(ucl,lcl))
}
if(chart=="R" && side=="U" && para=="yes")
    {
    ucl=sd*ru
    return(ucl)
    }
if(chart=="R" && side=="U" && para=="no")
    {
    ucl=(mean(gen4)/ud)*ru
    return(ucl)
    }
if(chart=="R" && side=="L" && para=="yes")
    {
   lcl=sd*rl
   return(lcl)
    }
if(chart=="R" && side=="L" && para=="no")
    {
      lcl=(mean(gen4)/ud)*rl
   return(lcl)
    }
if(chart=="R" && side=="T" && para=="yes")
    {
  ucl=sd*rtu
  lcl=sd*rtl
  return(c(ucl,lcl))
    }
if(chart=="R" && side=="T" && para=="no")
   {
  ucl=(mean(gen4)/ud)*rtu
  lcl=(mean(gen4)/ud)*rtl
  return(c(ucl,lcl))
    }
         }
#end of Function to compute control limits

gen=NULL
st=NULL
```

 # Function to detect out of control signals for different charts

```
scc=function(data,chart,r,m,alpha,side,sd,mean,para)
{
h=controllimt(data,chart,r,m,alpha,side,sd,mean,para)# Calling control limits function
 n=length(data[1,])
for(i in 1:length(data[,1]))
{
 if(chart=="Xbar")
      {
      gen[i]=rowMeans(data[i,])
      }
if(chart=="S")
      {
     gen[i]=sqrt(var(t(data[i,])))
      }
if(chart=="S2")
      {
      gen[i]=(var(t(data[i,])))
      }
if(chart=="R")
        {
        gen[i]=max(t(data[i,]))-min(t(data[i,]))
        }
  }

k=0 # intilization of counter
```

```
tc=length(data[,1]) # total# of subgroups
for(j in 1:(tc-m+1))    #implementation of different r/m rules on data to identify out of control points
{
 sum=0;
  for(i in 1:m)
   {
    if(side=="U")
              {
    if(gen[j]>h)
     {
      sum=sum+1
     }
     }
if(side=="L")
 {
  if(gen[j]<h)
   {
  sum=sum+1
   }
   }
if(side=="T")
  {
  if(gen[j]>h[1]||gen[j]<h[2])
  {
  sum=sum+1
  }
  }
 j=j+1
  }
if(sum>=r)
{
 k=k+1
 st[k]=j-1
 }
 }
                 # Displaying the final output
z=NULL
goo=NULL
if(length(st)>0)
{
 goo=st
}
else
{
goo="Nil"
}
z=length(goo)

cat("\n","\n","\t","\t","\t","SUMMARY FOR OUT-OF-CONTROL SIGNALS","\n","\t","\t","control
chart",":","\t","\t","\t","Shewart",chart,"\n","\t","\t","control limits","(",side,
")",":","\t","\t",controllimt(data,chart,r,m,alpha,side,sd,mean,para),"\n","\t","\t","False Alarm Rate",":","\t", "\t","\t",alpha,"\n",
"\t","\t","sample size",":","\t","\t","\t","\t",  n, "\n" ,"\t", "\t","subgroup size" ,":","\t","\t","\t", tc , "\n" ,"\t", "\t","Out of
control signals received at subgroups#",":", goo , "\n","\t","\t","Total# of out of control signals using",r,"/",m,":" ,length(st),"\n" )
# End of Displaying the final output
}
# End of Function to detect out of control signals for different charts
```

3. Description of the Code

In this section we provide a detailed description of our code in terms of its development and functionality for the desired purposes. This description will help practitioners to implement it easily for their desired use.

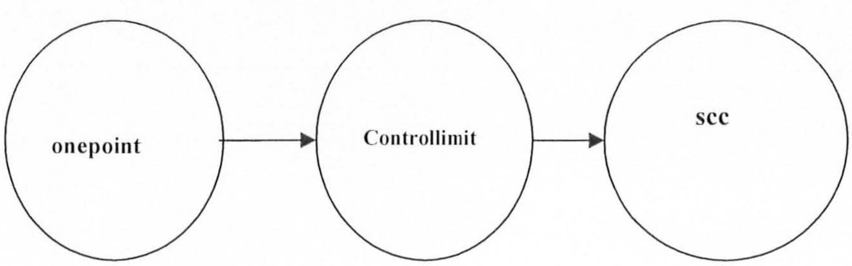
This code mainly consists of three user defined (our own developed) functions namely "onepoint", "controllimit" and "scc" to meet the desired objectives. Now we describe all the three functions one by one.

The function **"onepoint"** (**onepoint=function(alpha, r, m)**) provides the probability of a single point falling outside the respective signaling limits. It returns the optimal solution of the equation (1) by applying the index search approach using the arguments "alpha", "r"and "m". The argument "alpha" is pre-specified false alarm rate and it can 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"** (**controllimt=function(data,chart,r,m,alpha,side,sd,mean,para)**) returns the signaling (control) limit(s) for a given r/m rule. It depends upon the arguments "data" ,"chart", " r", "m" "alpha", "side", "sd", "mean", and "para". The argument "data" provides data to the function after attaching the specified data file from a given drive in computer. The argument "chart" can take any 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 any one of the three possible options namely "L" for lower sided, "U" for upper sided and "T" for two sided limits. The argument "para" refers to the parameter value and it can take either "yes" or "no" which means that parameter value is specified or not respectively. The arguments "sd" and "mean" represent the specified values of standard deviation and mean respectively and are used in the control limits construction only if "para" gets the value "yes" otherwise these are estimated from the attached data file. The other arguments are as defined earlier. It is to be noted that the function "**onepoint**" is called by the function "**controllimit**" to use the probability of a single point and setting the control limits.

The function **"scc"** (**scc=function(data,chart,r,m,alpha,side,sd,mean,para)**) is the front end function for user/practitioner. The arguments used in this function are the "data", "chart", "r", "m", "alpha", "side", "mean", and "para". All the arguments used in this function are as defined earlier and it calls the function "**controllimit**" to use the control limits. This function obtains the corresponding statistic values according to the choice of "chart" and then these statistics are consecutively compared with the values of control limit obtained form the function "**controllimit**". For those cases where points fall outside the control (signaling) limits the sample (subgroup) number is stored into a variable which are infact the out-of-control signals. The function **"scc"** 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 out-of-control signals. This summary output works as an attractive alternative to the graphical display of a control chart.

To further clarify the flow of our code we provide here a flow diagram:



4. Execution Procedure of the Code

In this section we list down the necessary steps which will help the practitioners to use our developed code for their data sets in order to receive out-of-control signals for their process characteristic(s) of interest. It is to be mentioned that the code works for the sample sizes from 2 to 25, which cover the generally considered choices used in practice according to the American standards of quality.

i) Enter the data in any data sheet (preferably in Excel) row wise for each sample (i.e. each row refers to a single sample of any size) and then save it as a file in the text format (e.g. abc.txt) with the type "Text (Tab delimited)" (cf. Verzani (2005) and Venables et al. (2010)) in any drive (e.g. f drive).

ii) Run the code of Section 2 in $\mathbb{R}$ editor or console and save the workspace so that it may be loaded before using the **"scc"** function next time (or alternatively **"scc"** may be added in the $\mathbb{R}$ library as a function of quality control package).

iii) Read the saved data file (as saved in (i)) in $\mathbb{R}$ by executing the following commands in $\mathbb{R}$ editor or console:

```
xyz=NULL
xyzc<- read.table("drivename:\\filename .txt",header=T)
```

where header =T means that the first row of the data file contains variable names in different columns (alternatively we may write header =F which means that data columns will be used without variable names). In these commands drivename and filename refer to the drive and file as saved in (i) (i.e. to give the path of the data file). Finally the data file will be stored in the dataframe of $\mathbb{R}$ with a particular name, say xyz. For example to attach the file 'abc' saved in f drive we will run the following statements:

```
xyz=NULL
xyz<- read.table("f:\\abc.txt",header=F)
```

iv) Attach the above read data frame, e.g. xyz, using the **attach** command as follows:

```
attach(xyz)
```

v) At last execute the following function **"scc"** after giving the desired options/values to the arguments used in it

```
scc(data=?,chart="?",r=?,m=?,alpha=?,side="?",sd=?,mean=?,para="?")
```

where each '?' refers to a specific choice for that particular argument e.g.

```
scc(data=xyz,chart="Xbar",r=1,m=1,alpha=0.0027,side="T",sd=1,mean=0,para="no")
```

is for $\bar{X}$ chart using 1/1 rule at α =0.0027 using two sided control limits when the standards (parameters) are not known (unspecified). The other options for these arguments may easily be entered following the description of Section 3 as per requirements of the practitioner.

vi) After the execution of **"scc"** function we will get the final output of the data which practitioner wants to test in the form of summary display in R-console providing the information of control chart type, control limits type (one or two sided), sample size, alpha, sample number where process is getting out-of-control, total number of samples the process gets out-of- control. This summary display will be easier for the practitioner to interpret out-of-control signals as compared to a graphical display of control chart.

5. Illustrative Example

To illustrate the application of our developed code for practitioner's convenience we consider a dataset taken from Alwan (2000) which refers back to W.A. Shewhart's

(1931) containing the data on 204 consecutive measurements of the electrical resistance of insulation in megohms. This data set may be seen on page 380 of Alwan (2000). We apply our code on the same data set using few of the runs rules schemes and see what output display is shown by our code.

For 1/1 rule on $\overline{X}$ chart we upload the code of Section 2 in $\mathbb{R}$ console and read & attach the said data file followed by executing the following statement:

```
scc(data=xyz,chart="Xbar",r=1,m=1,alpha=0.0027,side="T",sd=1,mean=0,para="no")
```

which gives the final output in $\mathbb{R}$ console as:

```
SUMMARY FOR OUT-OF-CONTROL SIGNALS
control chart :                Shewart Xbar
control limits ( T ) :          4977.989 4018.364
False Alarm Rate :                0.0027
sample size :                  4
subgroup size :                 51
Out of control signals received at subgroups# : 3 4 5 15 16 22 31 36 44 51
Total# of out of control signals using 1 / 1 : 10
```

This output summary is exactly in accordance with the results of graphical display given in Alwan (2000, page 381).

For 1/2 rule on $\overline{X}$ chart we run the following statement:

```
scc(data=xyz,chart="Xbar",r=1,m=2,alpha=0.0027,side="T",sd=1,mean=0,para="no")
```

which gives the final output in $\mathbb{R}$ console as:

```
SUMMARY FOR OUT-OF-CONTROL SIGNALS
control chart :                Shewart Xbar
control limits ( T ) :          5010.461 3985.892
False Alarm Rate :                0.0027
sample size :                  4
subgroup size :                 51
Out of control signals received at subgroups# : 3 4 5 6 22 23 31 32 36 37 44 45 51
Total# of out of control signals using 1 / 2 : 13
```

For 3/5 rule on S chart we run the following statement:

```
scc(data=xyz,chart="S",r=3,m=5,alpha=0.01,side="T",sd=1,mean=0,para="no")
```

which gives the final output in $\mathbb{R}$ console as:

```
SUMMARY FOR OUT-OF-CONTROL SIGNALS
 control chart :                Shewart S
 control limits ( T ) :          912.9824 4.64327
 False Alarm Rate :                0.01
 sample size :                  4
 subgroup size :                 51
 Out of control signals received at subgroups# : Nil
 Total# of out of control signals using 3 / 5 : 0
```

Similarly any option may be entered in the **"scc"** function to detect out-o-control signals for any chart at any false alarm rate with a given sample size and number of subgroups.

6. Summary, Conclusion, Recommendations and Future Proposals

It has never been an easy task to fix the false alarm rate at any level with different runs rules scheme to identify special causes. The existing packages/softwares generally work with few sensitizing rules/runs rules schemes and at very limited choices of false alarm rates. We have developed a functional code which can easily do it for many rules/schemes and for any choice of false alarm rate (this we have done for the four charts under discussion namely S, R, $\overline{X}$, S^2 charts at the moment and it may be done for other charts as well (one of our future projects)). Our code provides an attractive summary display of out-of-control signals in $\mathbb{R}$ console directly (replacing the graphical display which may not more smart choice for a practitioner). This code may be easily modified for the distributions other than normal. Moreover it may be extended for the EWMA and CUSUM charts (one of our future projects).

Keeping in view the above mentioned features of our proposed functional code we suggest its inclusion in the library of $\mathbb{R}$ language as a quality control tool/package with the name **"scc"** just like **"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 sensitive processes which have direct relation to the health care and engineering where correct and timely out-of –control signals may be of more concern (cf. Bonetti et al. (2000)).

References

Alwan, L. C. (2000). Statistical Process Analysis. McGraw-Hill International Editions: Singapore.

Antzoulakos, D. L. and Rakitzi, A. C. (2008) The modified r out of m control chart. Communication in Statistics-Simulations and Computations, 37, 396-408.

Bonetti, P. O., Waeckerlin, A., Schuepfer, G. and Frutiger, A. (2000). Improving time-sensitive processes in the intensive care unit: The example of 'Door-to-Needle Time' in acute myocardial infarction. *International Journal for Quality in Health Car*, 12, 311-317.

Khoo, M. B. C. (2004). Design of runs rules schemes. Quality Engineering, 16, 27-4

Klein, M. (2000). Two Alternatives to the Shewhart X control chart. Journal of Quality Technology; 32, 427-431.

Koutras, M. V., Bersimis, S. and Maravelakis, P. E. (2007). Statistical process control using Shewhart control chart with supplementary runs rules. Methodology and Computing in Applied Probability, 9, 207-224.

Montgomery, D. C. (2005). Introduction o Statisyical Quality Control. John Wiley and Sons, New York (5th Edition).

Murreell, P. and Gardiner, S. (2009). Quality control for statistical graphics: The graphicsQC package for R. Journal of Statistical Software, 30, 1-28.

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

Shewhart, W. (1931). Economic control of quality manufactured product, D. Van Nostrand, New York; reprinted by the American Society for Quality Control in 1980, Milwauker, WI.

Chapter 5

Summary and Conclusions

Shewart's type control charts are used to check whether process is in-control or out-of-control but they have the features to perform well for larger shifts. To overcome this problem different author have proposed sensitizing rules to increase the efficiency for small and moderate shifts. There are a number of control charting rules used with different control charts to decide between the above mentioned two states of control i.e. in-control and out-of-control. Some issues with these rules are highlighted in this study. By redefining and listing a set of rules we have evaluated their performance on the S^2, $\bar{X}$, R and S charts. We have compared the performance of these rules using their power curves to figure out the superior ones. Application of few of these rules with the real datasets is also shown to highlight their detection ability and use for practitioners. These extra sensitizing runs rules schemes provide a good support to their design structures but at the same time they introduce some problems with their properties. These problems include: biasedness and non-monotonicity in case of separate use of each rule in general and hence no independent identity of any rule for different types of shifts; need of simultaneous application of more rules at a time which results into an inflated false alarm rate and unattractive structures for practitioners use. By appropriately redefining these rules/schemes we have shown that each rule may have its own independent identification and hence may be used in its own capacity separately giving simplicity to the design structures of control charts and also overcoming the issues of inflated false alarm rate, biasedness and non-monotonicity.

Also we have developed a power computation code in $\mathbb{R}$ language namely **"power"** which will provide an ease to researchers in designing their projects and doing their further studies using different existing and newly introduced sensitizing rules and runs rules schemes designed for varying Shewhart's control charting structures. This code will provide help to researcher to compute the power for different options of r/m rules/schemes. The said code is flexible to be applied for any sample size, any false alarm rate, any type of control limits(one or two sided) and any amount of shift in process parameters for the four most commonly used Shewhart's type control charts namely S, R, $\bar{X}$, S^2 charts in case of both the specified and unspecified parameters cases. These mentioned benefits of our developed functional code are partially found in the features of the existing soft wares/packages and these may be enhanced by adding the features of our developed code as a function in their libraries dealing with quality control charting.

Moreover we have developed one more functional code in $\mathbb{R}$ language with the name **"scc"** which will provide an ease to practitioners in applying different existing and newly introduced sensitizing rules and runs rules schemes designed for varying Shewhart's control charting structures. This code will help identifying out-of-control signals very easily using different r/m rules. The said code is flexible to be applied for any number of subgroups, any sample size, any false alarm rate and any type of control limits(one or two sided) for the four most commonly used control charts namely S, R, $\bar{X}$, S^2 charts in case of both the specified and unspecified parameters cases. These mentioned benefits of our developed functional code are partially found in the features of the existing softwares/packages.