SUPPLEMENTARY MATERIAL TECHNICAL APPENDIX
Article: Comparison of control charts for monitoring clinical performance using binary data

APPENDIX TABLES

Table A1 – Comparison of out-of-control ARLs, in number of observations, for EWMA chart with different values of the weight lambda (λ)

Baseline event rate	Absolute shift	Odds ratio	In-control ARL	Different values of weight λ				
				0.10	0.05	0.01	0.005	0.001
1%	-0.5%	0.5	14,400	N/A#	N/A#	N/A#	N/A#	N/A#
	+1%	2.0		2,110	1,800	990	870	1,340
	+2%	3.1		720	580	350	350	700
10%	-5%	0.5	14,600	N/A#	N/A#	520	860	1,660
	+5%	1.6	9,800	1,030	650	400	490	1,240
	+10%	2.3	9,800	250	250	170	250	750
50%	-15%	0.5	8,750	240	170	250	400	1,560
	+15%	1.9		240	170	250	400	1,560
	+25%	3.0		70	70	160	280	1,130

Lower limit of chart is at zero so decreases cannot be detected

Table A2 – Comparison of out-of-control ARLs, in number of observations, for the CUSUM chart set up to detect increases and decreases in rates expressed as odds ratios

Baseline event rate	Absolute shift	Odds ratio	In-control ARL	Odds ratio (decreases)		Odds ratio (for increases in rates)			
				0.2	0.5	1.5	2	3	5
1%	-0.5%	0.5	14,400	1,800	1,500	N/A#	N/A#	N/A#	N/A#
	+1%	2.0		N/A†	N/A†	780	750	800	880
	+2%	3.1		N/A†	N/A†	360	320	300	320
10%	-5%	0.5	9,800	300	270	N/A#	N/A#	N/A#	N/A#
	+5%	1.6		N/A†	N/A†	310	330	460	710
	+10%	2.3		N/A†	N/A†	80	70	60	70
50%	-15%	0.5	8,750	180	110	N/A#	N/A#	N/A#	N/A#
	+15%	1.9		N/A†	N/A†	120	110	130	180
	+25%	3.0		N/A†	N/A†	60	50	40	50

ARLs are not calculated for detecting increases in rates because chart is set up to detect decreases

† ARLs are not calculated for detecting decreases in rates because chart is set up to detect increases

A. ADDITIONAL METHODS

A1.1 Constructing the Shewhart p-chart

The time-frame for monitoring is split into reporting periods of equal duration, such as months or quarters. The proportion within each reporting period is plotted on the vertical axis against time on the horizontal axis. The target proportion is added to the chart as a horizontal line for the unadjusted Shewhart p-chart – we have used the baseline event rate as the target in the main paper which we assume is known (see below for risk-adjustment).

The control limits are set to be a chosen number of standard errors away from the target proportion. We have used 2 and 3 standard error limits, which are typically often referred to as 2 sigma and 3 sigma limits because the standard deviation, sigma, of grouped data is a standard error. The number of observations within each period can vary, in which case the width of the limits changes over time. The standard error of a proportion is:

$$SE = \sqrt{\left(\frac{p(1-p)}{n}\right)} \quad (1)$$

A1.2 Risk-adjusting the Shewhart P-chart

The Shewhart P-chart can be risk-adjusted for patient case-mix by replacing the flat-line target rate with the predicted risk for each reporting period that can vary with the characteristics of patients treated within that period. The predicted risk is typically obtained by Indirect Standardisation.[1] The predicted risk for each patient is estimated by fitting a logistic regression model to all patients in the population. These predicted risks are then summed over the patients at the provider during each reporting period to obtain the expected number of events for each reporting period. The adjusted risk for the reporting period is then the ratio of the observed to expected number of events for that period multiplied by the overall risk. The standard errors are based on the adjusted risk at each reporting period.

A2.1 Constructing the g-chart

The statistic monitored with the g-chart is the number of individual observations between events, such as the number of patients or procedures.[2] The closely related t-chart could be used to monitor the number of days between events. The target value is the expected number of observations between events,s, – again using the baseline mean in the main paper. This can be written as a function of the baseline event rate:

$$E(s) = \frac{1-p}{p} \quad (2)$$

The control limits of the g-chart where the baseline event rate p is known, L standard errors from the target are:

$$\text{Control limits} = \frac{1-p}{p} \pm L\sqrt{\frac{1-p}{p^2}} \quad (3)$$

Where the baseline event rate, p, is estimated, the limits are based on the historical average number of observations between events, $\overline{s}$:

$$\text{Control limits} = \overline{s} \pm L\sqrt{\overline{s}(\overline{s}+1)} \quad (4)$$

For low event rates, corresponding to high values of $\overline{s}$, probability-based control limits derived from percentiles of the geometric distribution can be used:

$$\text{Upper probability limit} = \frac{\ln(1-\alpha)}{\ln(1-p)} \quad (5)$$

$$\text{Lower probability limit} = \frac{\ln(\alpha)}{\ln(1-p)} \quad (4)$$

where α is the specified false alarm rate.

A3.1 Constructing the EWMA

The trace is a weighted average of the current observation, x_i, and the previous value of the trace, s_{i-1}, as follows:

$$s_i = \lambda x_i + (1-\lambda)s_{i-1} \quad (6)$$

Equivalently, this can be written in terms of all previous observations, as follows:

$$s_i = \lambda \sum_{j=0}^{i-1}(1-\lambda)^j x_{i-j} + (1-\lambda)^i s_0 \quad (7)$$

The asymptotic limits of the EWMA with target rate p, L standard errors from the target are:

$$\text{Control limits} = p \pm L\sqrt{\frac{p(1-p)\lambda}{2-\lambda}} \quad (8)$$

The limits have a characteristic C-shape at the start of monitoring as the number of observations included in the estimate increases. In practice, data collection will pre-date the start of monitoring

and therefore the flat-line asymptotic limits will be more relevant. For completeness the limits of the EWMA at the ith data point are:

$$\text{Control limits} = p \pm L\sqrt{\frac{p(1-p)\lambda[1-(1-\lambda)^{2i}]}{2-\lambda}}$$

A3.2 Risk-adjusting the EWMA

The EWMA can be risk-adjusted for patient case-mix using a similar approach to that of the Shewhart P-chart. The flat-line target rate is replaced with the predicted risk. The predicted risk is calculated for each patient or procedure, and the target is then the weighted average of these predicted risks. The formula to estimate the risk-adjusted target is equation 5 or 6 above, but with the current observation, x_i, replaced by the predicted risk of the current observation. See Cook et al for more details.[3] The target therefore varies with the characteristics of patients treated over time. The predicted risk is again typically obtained by Indirect Standardisation, described in A1.2.[1] and the predicted risk is used in the formula for the standard error of the trace.

A4.1 Constructing the CUSUM

The CUSUM trace updates for each individual binary outcome, x, as follows:

$$s_i = s_{i-1} + \log(p1|p0)\ if\ x_i = 1, \text{ and}$$

$$s_i = s_{i-1} + \log((1-p1)|(1-p0))\ if\ x_i = 0 \qquad (9)$$

where s is the value of the CUSUM trace, *x* is the individual binary outcome (1 for an event, 0 for a non-event), $p0$ is the specified target rate and $p1$ is the alternative rate following a shift. $p1$ can be calculated for a specified shift size. S_i resets to zero when the control limit is crossed.

The CUSUM trace can also be expressed in terms of the target rate p0 and the odds ratio. In the above example, the odds ratio (OR) is just over 2. For an odds ratio of 2, the CUSUM can be written as:

$$s_i = s_{i-1} + \log 2 - \log(1+p0)\ if\ x_i = 1, \text{ and}$$

$$s_i = s_{i-1} - \log(1+p0)\ if\ x_i = 0 \qquad (10)$$

More generally, the formulae for any OR are:

$$s_i = s_{i-1} + \log OR - \log(1 + p0(OR - 1))\ if\ x_i = 1, \text{ and}$$

$$s_i = s_{i-1} - \log(1 + p0(OR - 1))\ if\ x_i = 0 \quad (11)$$

To create a two-sided CUSUM, a second trace can be created in a lower panel, with a different alternative rate (p1) and test specified and each value multiplied by -1 so that the lower trace mirrors the upper trace.[4]

The control limits take absolute values, since the CUSUM trace is a log likelihood ratio and thus on a common scale. Values for control limits commonly range from -5 to +5 but can take more extreme values. Control limits to give a specified in-control ARL are given in a look-up table in the appendix.

A4.2 Risk-adjusting the CUSUM

The CUSUM can be risk-adjusted by altering the target rate for different cases to reflect their predicted outcome (risk). The formulae for the risk-adjusted CUSUM (adapted from [4]) is:

$$s_i = s_{i-1} + \log OR - \log(1 + p^*(OR - 1))\ if\ x_i = 1, \text{ and}$$

$$s_i = s_{i-1} - \log(1 + p^*(OR - 1))\ if\ x_i = 0 \quad (12)$$

where p^* is the estimated target rate for each patient based on their risk, which can again be obtained by Indirect Standardisation as described in section A1.2.

B ESTIMATING AVERAGE RUN LENGTHS

B1 ESTIMATING AVERAGE RUN LENGTHS OF THE G-CHART

The ARL, in numbers of patients or procedures, is 1 divided by the probability of exceeding the relevant control limit, multiplied by the average number of patients-between-events.[2] For detecting a decrease in rate (increase in observations between events), this can be written:

$$ARL = \frac{1}{1 - P(s \leq upper\ limit | p = p^*)} \times \left(\frac{1 - p^*}{p^*}\right) \quad (13)$$

Where p* is either the target proportion (baseline event rate) or the alternative rate, and P() is the cumulative probability for the geometric distribution.

B2 ESTIMATING AVERAGE RUN LENGTHS OF THE P-CHART WITH SUPPLEMENTARY RULES, THE EWMA AND CUSUM USING MARKOV CHAIN METHODS

For the EWMA and CUSUM, the continuous value of the trace is approximated into a discrete number of values, known as *states* in the terminology of Markov Chain methods.[4,6,7] The pattern of transitions between states, and the corresponding transition probabilities, can be fully specified because each future value of the trace depends only upon the current value of the trace and the next observation i.e. the Markov property holds. For each of the charts, a matrix containing a defined number of states and transitions between states is defined, and established methods can then be applied to estimate approximate average run lengths [4,6,7].

For the Shewhart p chart with additional rules, the ARLs are calculated exactly using Markov Chain methods. Regions of the control chart are defined according to the rules used for triggering an alarm. For the rules used in this paper the regions are defined as: outside the outer limits; between the inner and outer limits; and between the target and inner limits. The probability of the trace being in each of these regions is calculated from the binomial distribution using the volume per reporting period and the event rate. The Markov chain states are then defined in terms of sequences of regions in which the trace fell in the preceding reporting periods. Only a relatively small number of preceding reporting periods are relevant to the signaling rules. Certain sequences of regions cause the chart to signal. In this way a transition matrix is constructed, and established Markov Chain methods can then be applied to estimate approximate average run lengths [4,6,7].

Our chart comparisons are for unadjusted charts and previous studies of the risk-adjusted CUSUM have examined the effect of varying distributions of patient-risk on ARLs.[8,9]

1 Spiegelhalter DJ. Funnel plots for comparing institutional performance. Statist. Med. 2005; 24:1185–1202.

2 Benneyan JC. Number-between g-type statistical quality control charts for monitoring adverse events. Health Care Management Science 2001; 4: 305-318.

3 Cook DA, Coory M, Webster RA. Exponentially weighted moving average charts to compare observed and expected values for monitoring risk-adjusted hospital indicators. BMJ Qual Saf 2011;20:469-474.

4 Steiner SH, Cook RJ, Farewell VT, et al. Monitoring surgical performance using risk-adjusted cumulative sum charts. Biostatistics 2000; 1: 441-452.

5 Montgomery DC. Introduction to Statistical Quality Control, 6th Edition. 2009. New York: Wiley & Sons

6 Cox DR and Miller HD. The theory of stochastic processes, 1977. 1st Edition. Chapman and Hall/CRC.

7 Fu JC, Spiring FA, Xie H. On the average run lengths of quality control schemes using a Markov chain approach. Statistics & Probability Letters 2002; 56: 369-380.

8 Tian W, Sun H, Zhang X and Woodall WH. The impact of varying patient populations on the in-control performance of the risk-adjusted CUSUM chart. Int J Qual Health Care 2015; 27: 31-36.

9 Loke CK, Gan FF. Joint monitoring scheme for clinical failures and predisposed risks. Qual Technol Quant Manag 2012; 9: 3-21.

C. R-CODE FOR CONSTRUCTING THE CHARTS

The code given for constructing the charts is that used to construct the charts in Figure 1 for 90-day mortality after bowel cancer surgery. The dataset *boweldata.csv* contains the following variables:

admidate	date of admission
Year	year of admission
Month	Month of admission
Death	an indicator of a death (1 for a death and 0 for survival)

```
#########################################################################
# LOAD DATA

library(foreign)

mydata = read.csv("boweldata.csv", header = TRUE)

# Sort by date and attach
date <- as.Date(mydata$admidate, "%d%B%Y")
sorteddata = mydata[order(date),]
attach(sorteddata)
#########################################################################
```

C1. SHEWHART P-CHART

```
#########################################################################
# AGGREGATE DATA
MonthlyID <- (Year-2006)*12 + Month
MonthlyRate = aggregate(Death,by = list(MonthlyID),mean,na.rm = T)
MonthlyCount = aggregate(Surgery,by = list(MonthlyID),sum,na.rm = T)
#########################################################################

#########################################################################
# PARAMETERS FOR P-CHART (target rate, width of limits)
target = 0.3
L = 3
#########################################################################

#########################################################################
# CALCULATE LIMITS
llim = target - L*sqrt(target*(1-target)/MonthlyCount)
```

```
ulim = target + L*sqrt(target*(1-target)/MonthlyCount)
##############################################################################

##############################################################################
# PLOT P-CHART
plot(MonthlyRate ~ c(1:24), col = "black", type = "l", las = 2, lwd = 1.7, axes = F,
     xlim = c(0,25), ylim = c(0.1,0.7),cex = 0.8,
     xlab = "Time in months", ylab = "% died")
axis(side = 1, at = c(1:24))
axis(side = 2, at = seq(0.1,0.7,0.1), labels = c("10%","20%","30%","40%","50%","60%","70%"),
     las = 2, cex = 1.5)
abline(h = target, col = "red")
lines(llim ~ c(1:24), col = "red", type = "l")
lines(ulim ~ c(1:24), col = "red", type = "l")
title("Shewhart p-chart")
##############################################################################
```

C2. G-chart

```
##############################################################################
# Parameters for g-chart (target value, width of limits)

target_gchart = (1-0.3)/0.3
L = 8.2


##############################################################################


##############################################################################
# Calculate limits
ulim_gchart = target_gchart + 8.2*sqrt(target_gchart*(1 + target_gchart))
llim_gchart = max(target_gchart - 8.2*sqrt(target_gchart*(1 + target_gchart)),0)
##############################################################################

##############################################################################
# Calculate trace

x = seq(1:length(Death))[which(Death==1)]
p = 0
g = rep(NA, length(x))
for (i in 1:length(x)) {

  g[i] = x[i] - p - 1
  p = x[i]

}
##############################################################################

##############################################################################
# Plot g-chart
plot(g ~ x, col = "black", type = "l", las = 2, lwd = 1.7, axes = F,
     ylim = c(0,15),cex = 0.8,
     xlab = "death* number", ylab = "Numbers of procedures between deaths")
axis(side = 1, at = seq(1,700,10))
axis(side = 2, at = seq(0,13), las = 2, cex = 1.5)
abline(h = target_gchart, col = "red")
abline(h = llim_gchart, col = "red", lty = 5)
abline(h = ulim_gchart, col = "red", lty = 5)
title("A g-chart")
##############################################################################
```

C3. EWMA

```
##############################################################################
# Parameters for EWMA (target rate, lambda, width of limits)
target = 0.3
lambda = 0.01
L = 3.75
##############################################################################
```

```
#########################################################################
# Calculate limits
llim_EWMA = target - L*sqrt((target)*(1-target)*(lambda/(2-lambda)))
ulim_EWMA = target + L*sqrt((target)*(1-target)*(lambda/(2-lambda)))
#########################################################################

#########################################################################
# Calculate trace
z = Death[complete.cases(Death)]
x = target
EWMA = rep(target,length(z))

for (i in 1:length(z)) {
    EWMA[i] = (1-lambda)*x + lambda*z[i]
    x = EWMA[i]
  }
#########################################################################

#########################################################################
# Plot EWMA
plot(EWMA~c(1:length(EWMA)),xlab = "patient number", type = "l", ylab = "% died ", axes = F,
    ylim = c(0.1,0.7))
axis(side = 2, at = seq(0.1,0.7,0.1), labels = c("10%","20%","30%","40%","50%","60%","70%"),
    las = 2, cex = 1.5)
axis(side = 1, at = seq(1,700,10))
abline(h = llim_EWMA, col = "red")
abline(h = ulim_EWMA, col = "red")
title("EWMA")
#########################################################################
```

C4. CUSUM

```
#########################################################################
# Parameters for CUSUM (target rate, hypothesized rate, control limit)
target = 0.3
pa = 0.4
limit = 5.7
#########################################################################

#########################################################################
# Calculate trace
p0 = target
s = z*log(pa/p0) + (1-z)*log((1-pa)/(1-p0))

CUSUM = rep(NA,length(z))
x = 0

for (i in 1:length(z)) {
  CUSUM[i] = ifelse(x + s[i]<0,0,x+s[i])
  CUSUM[i] = ifelse(x + s[i]>=limit,0,CUSUM[i])
  x = CUSUM[i]
}
#########################################################################

#########################################################################
# Plot CUSUM
plot(CUSUM~c(1:length(CUSUM)),xlab = "patient number", type = "l", ylab = "CUSUM",las = 2,
    axes = F)
abline(h = limit-0.1, col = "red")
points(CUSUM[393] ~ c(393), cex = 1.5, pch = 21)
axis(side = 1, at = seq(1,700,10))
axis(side = 2, at = c(0:7), las = 2)
title("CUSUM")
#########################################################################
```

D. R-CODE FOR ESTIMATING AVERAGE RUN LENGTHS

D1. SHEWHART P-CHART ARLs

```
# Calculate ARL using Markov Chains
# In order to calculate ARL for a decrease in rate, convert the target event rate and
# alternative event rate into rates of non-events (1 - event rate). This R code can then be
# used to calculate the ARL for an increase in the rate of non-events.
#### RULES FOR TRIGGER: 1 >3SEs OR 2/3 >2SEs OR 8/8 > target ####

# Target rate
p0=0.01

# Monthly volume
mv=200

# No. SEs for limits =h  ONLY WORKS IF h3>h2>h0
h0=0
h2=2
h3=3

# Control limits and target
ul3sem=p0+h3*sqrt(p0*(1-p0)/mv)
ul2sem=p0+h2*sqrt(p0*(1-p0)/mv)
ul0sem=p0+h0*sqrt(p0*(1-p0)/mv)

# Find minimum no. events needed to cross each threshold
ntrig3m=ceiling(mv*ul3sem)
ntrig2m=ceiling(mv*ul2sem)
ntrig0m=ceiling(mv*ul0sem)

# Probability if being in each region
p0trig3m=pbinom(ntrig3m-1,mv,p0,lower.tail=F)
p0trig2m=pbinom(ntrig2m-1,mv,p0,lower.tail=F)
p0trig0m=pbinom(ntrig0m-1,mv,p0,lower.tail=F)

# Markov chain for p-chart with 2 additional rules
p01=1-p0trig0m
p02=p0trig0m-p0trig2m
p03=p0trig2m-p0trig3m
p04=p0trig3m

library(MASS)
####################################################
# Function to find ARL from transition matrix
solve = function(m) {
  r = nrow(m)-1
  m = ginv(diag(r)-m[1:r,1:r]) %*% rep(1,r)
  m[1]
}
####################################################

# define transition matrix
# dimension of matrix
g=22
# Transition matrix for in-control ARL
trans.matrix0 = matrix(0, nrow = g,ncol = g)
trans.matrix0[,g]<-rep(p04,g)
trans.matrix0[,1]<-rep(p01,g)
trans.matrix0[g,]<-rep(1,g)
trans.matrix0[1,2]<-p02
trans.matrix0[1,3]<-p03
trans.matrix0[2,4]<-p02
trans.matrix0[2,5]<-p03
trans.matrix0[3,1]<-0
trans.matrix0[3,6]<-p01
trans.matrix0[3,7]<-p02
trans.matrix0[3,22]<-p03+p04
trans.matrix0[4,8]<-p02
trans.matrix0[4,9]<-p03
trans.matrix0[5,1]<-0
trans.matrix0[5,6]<-p01
trans.matrix0[5,10]<-p02
trans.matrix0[5,22]<-p03+p04
```

```
trans.matrix0[6,2]<-p02
trans.matrix0[6,22]<-p03+p04
trans.matrix0[7,8]<-p02
trans.matrix0[7,22]<-p03+p04
trans.matrix0[8,11]<-p02
trans.matrix0[8,12]<-p03
trans.matrix0[9,1]<-0
trans.matrix0[9,6]<-p01
trans.matrix0[9,13]<-p02
trans.matrix0[9,22]<-p03+p04
trans.matrix0[10,11]<-p02
trans.matrix0[10,22]<-p03+p04
trans.matrix0[11,14]<-p02
trans.matrix0[11,15]<-p03
trans.matrix0[12,1]<-0
trans.matrix0[12,6]<-p01
trans.matrix0[12,16]<-p02
trans.matrix0[12,22]<-p03+p04
trans.matrix0[13,14]<-p02
trans.matrix0[13,22]<-p03+p04
trans.matrix0[14,17]<-p02
trans.matrix0[14,18]<-p03
trans.matrix0[15,1]<-0
trans.matrix0[15,6]<-p01
trans.matrix0[15,19]<-p02
trans.matrix0[15,22]<-p03+p04
trans.matrix0[16,17]<-p02
trans.matrix0[16,22]<-p03+p04
trans.matrix0[17,20]<-p02
trans.matrix0[17,21]<-p03
trans.matrix0[18,1]<-0
trans.matrix0[18,6]<-p01
trans.matrix0[18,20]<-p02
trans.matrix0[18,22]<-p03+p04
trans.matrix0[19,20]<-p02
trans.matrix0[19,22]<-p03+p04
trans.matrix0[20,22]<-p02+p03+p04
trans.matrix0[21,1]<-0
trans.matrix0[21,6]<-p01
trans.matrix0[21,22]<-p02+p03+p04

# ARL0
arl0e=solve(trans.matrix0)
arl0=ceiling(arl0e)
arl0

# ARL0 in procedures
arl0*mv
```

D2. G-CHART ARLs

```
############################################################
# Limits based on critical values
# The ARL calculations are for detecting decreases in rates (increases in observations
# between events).  These rely on the geometric probability of exceeding the upper limit.
# For detecting increases in rates (decreases in observations between events), this can be
# replaced with the geometric probability of exceeding the lower limit.

getlims = function(p0,k) {
  target = (1-p0)/p0
  llim = max(target - k*sqrt((1-p0)/p0^2),0)
  ulim = target + k*sqrt((1-p0)/p0^2)
  return(c(llim,ulim))
}

# estimates ARL measured in numbers of patients or procedures

ARL = function(p0,k,p) {
  llim = getlims(p0,k)[1]
  ulim = getlims(p0,k)[2]
  prob = 1 - pgeom(round(ulim),p)
  ARL_events =   1/(prob)
  s = (1-p)/p
  ARL = ARL_events*s
  return(ARL)
```

```
}

# Probability based limits
problims = function(p0,alpha) {
  target = (1-p0)/p0
  llim = max(log(1-alpha)/log(1-p0),0)
  ulim = log(alpha)/log(1-p0)
  return(c(llim,ulim))
  }

ARL = function(p0,alpha,p) {
  llim = problims(p0,alpha)[1]
  ulim = problims(p0,alpha)[2]
  prob = 1 - pgeom(round(ulim),p)
  ARL_events =   1/(prob)
  s = (1-p)/p
  ARL = ARL_events*s
  return(ARL)
}

############################################################
```

D3. EWMA ARLs

```
# In order to calculate ARL for a decrease in rate, convert the target event rate and
# alternative event rate into rates of non-events (1 - event rate). This R code can then be
# used to calculate the ARL for an increase in the rate of non-events.

############################################################
# 1. Choose settings for Markov chains, chart and data     #
# 2. Calculate control limits                              #
# 3. Define transition matrix                              #
# 4. Apply solve function                                  #
############################################################

library(MASS)

############################################################
# 1. Choose settings                                       #
                                                           #
# No. discrete states for Markov chain g                   #
g = 2000                                                   #
                                                           #
# Specify target rate p0, actual rate p1, weight lambda    #
# and control limits L SEs from target                     #
p0 = 0.20                                                  #
p1 = 0.40                                                  #
lambda = 0.01                                              #
L = 3.15                                                   #
############################################################

############################################################
# 2. Asymptotic upper limit of EWMA                        #
var.target = p0*(1-p0)                                     #
se.ewma = sqrt(var.target*lambda/(2-lambda))               #
h = p0 + L*se.ewma                                         #
############################################################


############################################################
# 3. Define transition matrix                              #
                                                           #
# Calculate multiplier to get required number of states    #
a = round(g/h)                                             #
                                                           #
# i is current state. jscore is state moving to            #
jscore = function(lambda,i,a,x) {                          #
  round((1-lambda)*i + lambda*a*x)                         #
}                                                          #
                                                           #
trans.matrix = function(p,g) {                             #
  A = matrix(0, nrow = g,ncol = g)                         #
  for (i in 1:g) {                                         #
    j = max(1,jscore(lambda,i,a,0))                        #
```

```
    k = min(g,jscore(lambda,i,a,1))                          #
    A[i,j] = 1-p                                             #
    A[i,k] = p                                               #
  }                                                          #
                                                             #
A[g,g-1]=0                                                   #
A[g,g] = 1                                                   #
}                                                            #
trans.matrix(p1,g)[g,]                                       #
#############################################################

#############################################################
# 4. Function to find ARL from transition matrix             #
solve = function(m) {                                        #
  r = nrow(m)-1                                              #
  m = ginv(diag(r)-m[1:r,1:r]) %*% rep(1,r)                  #
  m[1]                                                       #
}                                                            #
                                                             #
solve(trans.matrix(p1,g))                                    #
#############################################################
```

D4. CUSUM ARLs

```
#############################################################
# 1. Choose settings for Markov chains, chart and data      #
# 2. Calculate the score to add or subtract to the trace    #
# 3. Define transition matrix                               #
# 4. Apply solve function                                   #
#############################################################

library(MASS)

#############################################################
# Function to calculate alternative rate as a function of   #
# p0 and odds ratio (or)                                    #
#                                                           #
getp = function(p0,or) {                                     #
  odds0 = p0/(1-p0)                                          #
  odds1 = or*odds0                                           #
  p1 = odds1/(1 + odds1)                                     #
  return(p1)                                                 #
}                                                            #
#############################################################

#############################################################
# 1. Choose settings                                        #
                                                            #
# No. discrete states for Markov chain g                    #
g = 2000                                                     #
                                                             #
# Specify target rate p0, alternative rate pa,              #
# actual rate p1 and control limits h                       #
p0 = 0.20                                                    #
pa = getp(p0,2)                                              #
p1 = 0.40                                                    #
h = 3                                                        #
#############################################################

#############################################################
# 2. Calculate the score to add or subtract to the trace    #
                                                            #
score = function(x) {                                        #
  ifelse(x==1,log(pa/p0),log((1-pa)/(1-p0)))                 #
}                                                            #
#############################################################


#############################################################
# 3. Define transition matrix                               #
                                                            #
# Calculate multiplier to get required number of states     #
a = round(g/h)                                               #
```

```
                                                                    #
# Calculate shift size in terms of number of states                 #
# moved for given score (typically of value < 1)                    #
                                                                    #
iscore = function(x) {                                              #
  round(score(x)*a)                                                 #
}                                                                   #
                                                                    #
trans.matrix = function(p,g) {                                      #
  A = matrix(0, nrow = g,ncol = g)                                  #
  for (i in 1:g) {                                                  #
    j = max(1,i+iscore(0))                                          #
    k = min(g,i+iscore(1))                                          #
    A[i,j] = 1-p                                                    #
    A[i,k] = p                                                      #
  }                                                                 #
A[g,g-1]=0                                                          #
A[g,g] = 1                                                          #
}                                                                   #
                                                                    #
trans.matrix(p0,g)[g,]                                              #
#####################################################################


#####################################################################
# 4. Function to find ARL from transition matrix                    #
solve = function(m) {                                               #
  r = nrow(m)-1                                                     #
  m = ginv(diag(r)-m[1:r,1:r]) %*% rep(1,r)                         #
  m[1]                                                              #
}                                                                   #
                                                                    #
solve(trans.matrix(p1,g))                                           #
#####################################################################


#####################################################################
# 3a. Define transition matrix to detect reduction                  #
# where or<1 and pa<p0                                              #
#                                                                   #
trans.matrix2 = function(p,g) {                                     #
  A = matrix(0, nrow = g,ncol = g)                                  #
  for (i in 1:g) {                                                  #
    j = min(g,i+iscore(0))                                          #
    k = max(1,i+iscore(1))                                          #
    A[i,j] = 1-p                                                    #
    A[i,k] = p                                                      #
  }                                                                 #
A[g,g-1]=0                                                          #
A[g,g] = 1                                                          #
}                                                                   #
#####################################################################
```

E. IN-CONTROL ARLs FOR A RANGE OF TARGET RATES AND CONTROL LIMIT WIDTHS

EWMA

1% target	Lambda	
Limit width (no. SEs from target)	**0.01**	**0.05**
3.0	2,590	430
3.5	4,860	590
4.0	9,760	810
4.5	20,810	1,130
5.0	47,190	1,820

5% target	Lambda	
Limit width (no. SEs from target)	**0.01**	**0.05**
3.0	4,400	590
3.5	10,910	1,180
4.0	30,950	2,560
4.5	99,960	6,000
5.0	>100,000	15,220

10% target	Lambda	
Limit width (no. SEs from target)	**0.01**	**0.05**
3.0	5,510	810
3.5	15,580	1,940
4.0	52,710	5,180
4.5	>100,000	15,650

20% target	Lambda	
Limit width (no. SEs from target)	**0.01**	**0.05**
3.0	7,000	1,230
3.5	23,180	3,700
4.0	95,820	13,530
4.5	>100,000	61,680

40% target	Lambda	
Limit width (no. SEs from target)	**0.01**	**0.05**
3.0	9,780	2,380
3.2	16,050	4,260
3.5	38,430	11,310
4.0	>100,000	78,940

CUSUM

1% target	Shift size in odds ratios	
Limit height	**2**	**4**
3.0	7,850	1,700
3.5	14,290	2,760
4.0	25,560	4,800
4.5	45,140	7,880
5.0	78,960	13,150

5% target	Shift size in odds ratios	
Limit height	**1.5**	**2.5**
3.0	3,850	920
3.5	6,430	1,610
4.0	10,480	2,790
4.5	16,800	4,780
5.0	26,450	8,030

10% target	Shift size in odds ratios	
Limit height	**1.3**	**2**
3.0	5,050	710
3.5	9,300	1,260
4.0	16,780	2,140
4.5	29,900	3,620
5.0	52,700	6,060

20% target	Shift size in odds ratios	
Limit height	**1.3**	**1.9**
3.0	3,400	600
3.5	5,900	1,050
4.0	9,920	1,810
4.5	16,320	3,060
5.0	26,460	5,100

40% target	Shift size in odds ratios	
Limit height	**1.2**	**1.8**
3.0	3,870	490
3.5	6,940	870
4.0	12,190	1,480
4.5	21,100	2,500
5.0	36,100	4,170