Scientia Iranica

Supplementary information

Product Acceptance Determination Based on EWMA Yield Index Using Repetitive and MDS Sampling Schemes

Muhammad Aslam[a *], Muhammad Azam[b], Rehan Ahmad Khan Sherwani[c], Ching- Ho Yen[d] and Chi-Hyuck Jun[e]

a. *Department of Statistics, Faculty of Science, King Abdulaziz University, Jeddah 21551, Saudi Arabia, Email: aslam_ravian@hotmail.com; Mobile :00966593329841*
b. *Department of Statistics and Computer Sciences, University of Veterinary and Animals Sciences, Lahore 54000, Pakistan, Email: mazam72@yahoo.com*
c. *College of Statistical and Actuarial Sciences, University of the Punjab, Lahore, Pakistan, Email: rehan.stat@pu.edu.pk*
d. *Department of Industrial Engineering & Management Information , Huafan University, Taipei, Taiwan, Email: jimyen@cc.hfu.edu.tw*
e. *Department of Industrial and Management Engineering, POSTECH, Pohang 37673, Republic of Korea; Email: chjun@postech.ac.kr*

Program for the computation of results.

```
# Title: Product Acceptance Determination Based on EWMA Yield Index Using Repetitive and MDS Sampling Schemes

#######  Proposed Plan using Repetitive Sampling
EWMA_Yield <- function(j,nos,alpha,beta)                    # nums <- no of samples in each draw
        {
        kas = c(); krs = c(); ns = c(); lp1s = c(); lp2s = c(); ASN2min = c(); #j = 10
        #cp1 = 1.4; ca1 = 0.912325; ld = 0.1; nos = 1000; i = 2; sAQL = 1.33; sLTPD = 1.0; alpha = 0.05; beta = 0.1
for(i in 1:j){
        n = sample(2:50, nos, TRUE)
        ka <- runif(nos,0.8,2.5); kr = runif(nos,0.8,ka)
        c11 = 3*cp1*(2-ca1); c21 = 3*cp1*ca1
        a1  = (1/sqrt(2))*(c11*dnorm(c11)+c21*dnorm(c21))
        b1 = dnorm(c11) - dnorm(c21)
        spk = sAQL
        num11 = ka - spk; num21 = kr - spk; lds = ld/(2-ld); dnum1 = sqrt(lds*(a1^2 + b1^2)/(36*n*dnorm(3*spk)^2))
        P11 = (1 - pnorm(num11/dnum1)) + (pnorm(num21/dnum1) - pnorm(num11/dnum1))*(1 - pnorm(num11/dnum1))^i
        P21 = (pnorm(num11/dnum1) - pnorm(num21/dnum1))*(1 - (1 - pnorm(num11/dnum1))^i)
        spk = sLTPD
        #cp2 = 1.1; ca2 = 0.845651
        c12 = 3*cp2*(2-ca2); c22 = 3*cp2*ca2
        a2  = (1/sqrt(2))*(c12*dnorm(c12)+c22*dnorm(c22))
```

```
	b2 = dnorm(c12) - dnorm(c22)
	num12 = ka - spk; num22 = kr - spk; lds = ld/(2-ld); dnum2 = sqrt(lds*(a2^2 + b2^2)/(36*n*dnorm(3*spk)^2))
	P12 = (1 - pnorm(num12/dnum2)) + (pnorm(num22/dnum2) - pnorm(num12/dnum2))*(1 - pnorm(num12/dnum2))^i
	P22 = (pnorm(num12/dnum2) - pnorm(num22/dnum2))*(1 - (1 - pnorm(num12/dnum2))^i)
	pieA1 = P11/(1-P21); pieA2 = P12/(1-P22)
	ASN2 = n/(1-pnorm(num12/dnum2)+pnorm(num22/dnum2))
	w = which(pieA1 >= (1-alpha) & pieA2 <= beta)
	n = n[w]; ASN2 = ASN2[w]; ka = ka[w]; kr = kr[w]; lp1 = pieA1[w]; lp2 = pieA2[w]
	ASN2min[i] = min(ASN2); wASN2min = which.min(ASN2)
	kas[i] = ka[wASN2min]; krs[i] = kr[wASN2min]; ns[i] = n[wASN2min]; lp1s[i] = lp1[wASN2min]; lp2s[i] = lp2[wASN2min]
		}
	wASN2minall = which.min(ASN2min)				# which of ASN2 is min out of all samples
	kas = kas[wASN2minall]; krs = krs[wASN2minall]; ns = ns[wASN2minall]; lp1s = lp1s[wASN2minall]; lp2s = lp2s[wASN2minall]
	ASN2min = ASN2min[wASN2minall]
	final_output = cbind(ns,kas,krs,ASN2min,lp1s,lp2s)
	print(final_output)
		}
nos = 10000; i = 2; j = 500 ; ld = 0.1
cp1 = 1.4; cp2 = 1.1; ca1 = 0.912325; ca2 = 0.845651; sAQL = 1.33; sLTPD = 1.0
#cp1 = 1.6; cp2 = 1.4; ca1 = 0.906850; ca2 = 0.912325; sAQL = 1.50; sLTPD = 1.33
alpha = 0.01
EWMA_Yield(j,nos,alpha,0.01)

############  Proposed Plan using Lot Resubmitted Sampling
EWMA_Yield <- function(j,nos,alpha,beta)			# nums <- no of samples in each draw
	{
	kas = c(); krs = c(); ns = c(); lp1s = c(); lp2s = c(); ASN2min = c(); #j = 10
	#cp1 = 1.4; ca1 = 0.912325; ld = 0.1; nos = 1000; i = 2; sAQL = 1.33; sLTPD = 1.0; alpha = 0.05; beta = 0.1
for(i in 1:j){
	n = sample(2:500, nos, TRUE)
	ka <- runif(nos,0.8,4.5); kr = runif(nos,0.8,ka)
	c11 = 3*cp1*(2-ca1); c21 = 3*cp1*ca1
	a1  = (1/sqrt(2))*(c11*dnorm(c11)+c21*dnorm(c21))
	b1 = dnorm(c11) - dnorm(c21)
	spk = sAQL
	num11 = ka - spk; num21 = kr - spk; lds = ld/(2-ld); dnum1 = sqrt(lds*(a1^2 + b1^2)/(36*n*dnorm(3*spk)^2))
	P11 = 1 - pnorm(num11/dnum1)
	P21 = pnorm(num11/dnum1) - pnorm(num21/dnum1)
	spk = sLTPD
	#cp2 = 1.1; ca2 = 0.845651
	c12 = 3*cp2*(2-ca2); c22 = 3*cp2*ca2
	a2  = (1/sqrt(2))*(c12*dnorm(c12)+c22*dnorm(c22))
	b2 = dnorm(c12) - dnorm(c22)
	num12 = ka - spk; num22 = kr - spk; lds = ld/(2-ld); dnum2 = sqrt(lds*(a2^2 + b2^2)/(36*n*dnorm(3*spk)^2))
	P12 = 1 - pnorm(num12/dnum2)
	P22 = pnorm(num12/dnum2) - pnorm(num22/dnum2)
	pieA1 = 1 - (1 - P11)^i; pieA2 = 1 - (1 - P12)^i
	ASN2 = n*(1 - (1 - P12)^i)/P12
	w = which(pieA1 >= (1-alpha) & pieA2 <= beta)
```

```
        n = n[w]; ASN2 = ASN2[w]; ka = ka[w]; kr = kr[w]; lp1 = pieA1[w]; lp2 = pieA2[w]
        ASN2min[i] = min(ASN2); wASN2min = which.min(ASN2)
        kas[i] = ka[wASN2min]; krs[i] = kr[wASN2min]; ns[i] = n[wASN2min]; lp1s[i] = lp1[wASN2min]; lp2s[i]
= lp2[wASN2min]
                }
        wASN2minall = which.min(ASN2min)                        # which of ASN2 is min out of all samples
        kas = kas[wASN2minall]; krs = krs[wASN2minall]; ns = ns[wASN2minall]; lp1s = lp1s[wASN2minall];
lp2s = lp2s[wASN2minall]
        ASN2min = ASN2min[wASN2minall]
        final_output = cbind(ns,kas,krs,ASN2min,lp1s,lp2s)
        print(final_output)
                }
nos = 10000; j = 100 ; ld = 0.1; i = 2
cp1 = 1.4; cp2 = 1.1; ca1 = 0.912325; ca2 = 0.845651; sAQL = 1.33; sLTPD = 1.0
#cp1 = 1.6; cp2 = 1.4; ca1 = 0.906850; ca2 = 0.912325; sAQL = 1.50; sLTPD = 1.33
alpha = 0.01
EWMA_Yield(j,nos,alpha,0.01)
#######  Designing of Proposed Plan using Multiple Dependent State Sampling
EWMA_Yield <- function(j,nos,alpha,beta)                        # nums <- no of samples in each draw
        {
        kas = c(); krs = c(); ns = c(); lp1s = c(); lp2s = c(); nmin = c(); #j = 10
        #cp1 = 1.4; ca1 = 0.912325; ld = 0.1; nos = 1000; i = 2; sAQL = 1.33; sLTPD = 1.0; alpha = 0.05; beta =
0.1
for(i in 1:j){
        n = sample(2:1000, nos, TRUE)
        ka <- runif(nos,0.2,3.5); kr = runif(nos,0.2,ka)
        c11 = 3*cp1*(2-ca1); c21 = 3*cp1*ca1
        a1  = (1/sqrt(2))*(c11*dnorm(c11)+c21*dnorm(c21))
        b1 = dnorm(c11) - dnorm(c21)
        spk = sAQL
        num11 = ka - spk; num21 = kr - spk; lds = ld/(2-ld); dnum1 = sqrt(lds*(a1^2 +
b1^2)/(36*n*dnorm(3*spk)^2))
        pieA1 = (1 - pnorm(num11/dnum1)) + (pnorm(num21/dnum1) - pnorm(num11/dnum1))*(1 -
pnorm(num11/dnum1))^i
        spk = sLTPD
        #cp2 = 1.1; ca2 = 0.845651
        c12 = 3*cp2*(2-ca2); c22 = 3*cp2*ca2
        a2  = (1/sqrt(2))*(c12*dnorm(c12)+c22*dnorm(c22))
        b2 = dnorm(c12) - dnorm(c22)
        num12 = ka - spk; num22 = kr - spk; lds = ld/(2-ld); dnum2 = sqrt(lds*(a2^2 +
b2^2)/(36*n*dnorm(3*spk)^2))
        pieA2 = (1 - pnorm(num12/dnum2)) + (pnorm(num22/dnum2) - pnorm(num12/dnum2))*(1 -
pnorm(num12/dnum2))^i
        w = which(pieA1 >= (1-alpha) & pieA2 <= beta)
        n = n[w]; ka = ka[w]; kr = kr[w]; lp1 = pieA1[w]; lp2 = pieA2[w]
        nmin[i] = min(n); wnmin = which.min(n)
        kas[i] = ka[wnmin]; krs[i] = kr[wnmin]; ns[i] = n[wnmin]; lp1s[i] = lp1[wnmin]; lp2s[i] = lp2[wnmin]
                }
        wnminall = which.min(nmin)                              # which of ASN2 is min out of all samples
        kas = kas[wnminall]; krs = krs[wnminall]; ns = ns[wnminall]; lp1s = lp1s[wnminall]; lp2s =
lp2s[wnminall]
        final_output = cbind(ns,kas,krs,lp1s,lp2s)
        print(final_output)
                }
nos = 10000; i = 2; j = 1000 ; ld = 0.1
#cp1 = 1.4; cp2 = 1.1; ca1 = 0.912325; ca2 = 0.845651; sAQL = 1.33; sLTPD = 1.0
```

cp1 = 1.6; cp2 = 1.4; ca1 = 0.906850; ca2 = 0.912325; sAQL = 1.50; sLTPD = 1.33
alpha = 0.01
EWMA_Yield(j,nos,alpha,0.01)