Control Estadístico de Calidad

Entrega 1. 2024-II

Michel Mendivenson Barragán Zabala
Departamento de Estadística
Universidad Nacional de Colombia

mbarraganz@unal.edu.co

Juan Sebastián Huertas Pirajan
Departamento de Estadística
Universidad Nacional de Colombia

juhuertasp@unal.edu.co

Diego Andres Paez Molina
Departamento de Estadística
Universidad Nacional de Colombia

dpaezm@unal.edu.co

22 de marzo de 2024

Ejercicio 1: Sea $X \sim N(\mu, \sigma)$ una característica de calidad. Mediante simulaciones, establezca el comportamiento del ARL (en control y fuera de él) de las Cartas R y S para observaciones normales con límites 3σ y muestras de tamaño (a) $n = 3$ y (b) $n = 10$ ¿Qué regularidades observa?

> Para la implementación de la solución, se creará en R una función que nos permita simular cuantás veces querramos el momento en que un proceso da una alerta (bien sea verdadera o falsa) con argumentos que nos permitan modificar tanto el tamaño de muestra n como los límites de la carta de control y su línea central para cada una de las cartas. Las funciones se definen como sigue:

```
RunLengthS = function(mu = 0, sigma = 1, CorrimientoSigma = 1,n = 3, m = 1000){
  S = c() ; c4 = sqrt(2/(n-1)) * gamma(n/2) / gamma((n-1)/2); CLs = c4 * sigma;
  LCL = sigma * (c4 - 3 * sqrt(1 - c4**2)); UCL = sigma * (c4 + 3 * sqrt(1 - c4**2))

  pb = txtProgressBar(min = 0, max = m, style = 3) # Barra de progreso

  i = 0
  while (length(S) < m){
    s = sd(rnorm(n, mu, CorrimientoSigma))
    i = i + 1
    if (s < LCL | s > UCL){
      S = c(S, i)
      i  = 0; setTxtProgressBar(pb, length(S))
    }}
  return(S)}

RunLengthR = function(mu = 0, sigma = 1, CorrimientoSigma = 1,n = 3, m = 1000){
  # Constantes carta R
  d3 = c(0.853, 0.888, 0.880, 0.864, 0.848, 0.833, 0.820, 0.808, 0.797, 0.787, 0.770,
         0.763, 0.756, 0.750, 0.744, 0.739, 0.734, 0.729, 0.724, 0.72, 0.716, 0.712,0.708)
  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, 3.819, 3.858, 3.895, 3.931)
  d3 = d3[n-1]; d2 = d2[n-1]; R = c(); UCL = (d2 + 3 * d3) * sigma; LCL = (d2 - 3 * d3) * sigma

  pb = txtProgressBar(min = 0, max = m, style = 3) # Barra de progreso
```

```
i = 0
while(length(R) < m){
  r = diff(range(rnorm(n, mu, CorrimientoSigma)))
  i = i + 1
  if (r < LCL | r > UCL){
    R = c(R, i)
    i = 0; setTxtProgressBar(pb, length(R))
  }}
return(R)}
```

Tenga en cuenta que la salida de la función es un vector con los valores de los tiempos en que se detecto una señal dados los límites y la línea central correspondientes al proceso en control (El proceso en control se definió con $\mu = 0$, $\sigma = 1$ y además para cada uno se tomaron m $= 1000$ muestras de tiempos en que se generó una alerta). Los resultados para diferentes corrimientos se grafican a continuación:

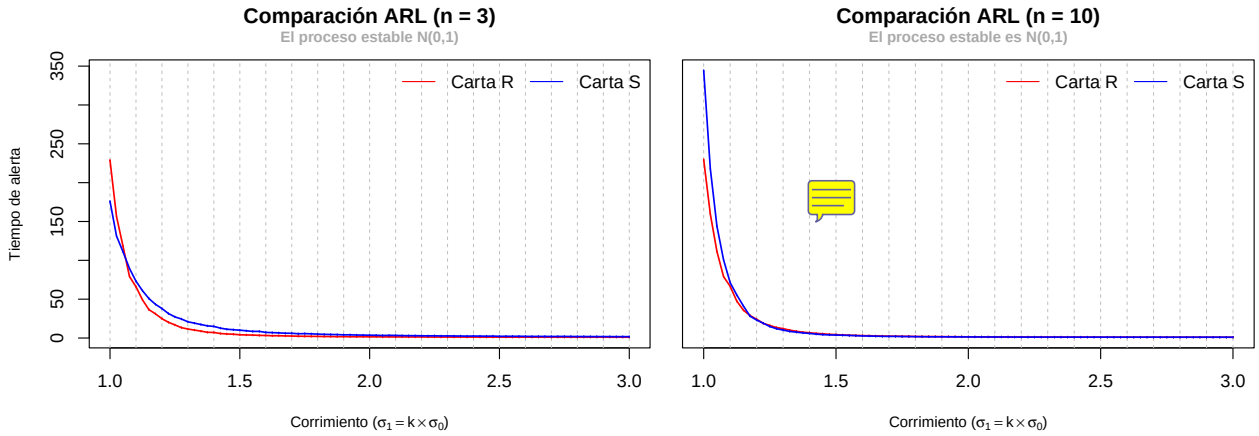

Figura 1: Comparación de ARL entre cartas R y S para n $= 3, 10$.

Se tomaron 81 corrimientos a intervalos regulares desde 1 hasta 3 por lo que una tabla no sería útil para analizar la información. Para $n = 3$ a la carta R le toma, en promedio, más tiempo para producir falsas alarmas para un proceso estable mientras que para los corrimientos de $1{,}1 \times \sigma$ en adelante, la carta R también tiende a tardar menos en dar una alarma verdadera. Por otro lado, la carta S para un tamaño de subgrupos racionales de 10 tarda en promedio más tiempo que la carta R para dar falsas alarmas y en cuanto a alarmas verdaderas no parece diferenciarse demasiado respecto a la carta R.

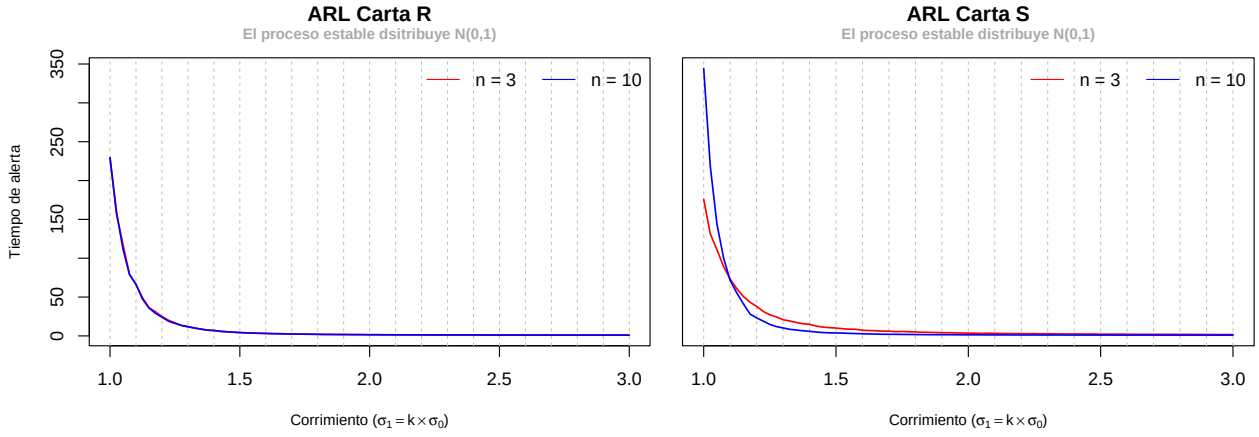

Figura 2: Comparación de ARL para cada carta con n $= 3, 10$

Además vemos que la carta R tiene un comportamiento más estable respecto al cambio del tamaño de los subgrupos racionales mientras que S tiende a comportarse mejor a medida que este aumenta debido a, en promedio, presentar tanto una menor frecuencia de falsas alarmas como mayor frecuencia de alarmas verdaderas.

Ejercicio 2: Sea $X \sim N(\mu, \sigma)$ una característica de calidad. Se sabe que los valores objetivo de los parámetros del proceso son $\mu = \mu_0$ y $\sigma = \sigma_0$. Construir las curvas OC de la carta S^2 con límites de probabilidad. Interpretar los resultados

Recordemos que los límites de probabilidad de una carta de control S^2 se calculan usando la distribución chi-cuadrada ($UCL = \frac{\sigma_0^2}{n-1}\chi^2_{(1-\alpha/2;\ n-1)}$ y $LCL = \frac{\sigma_0^2}{n-1}\chi^2_{(\alpha/2;\ n-1)}$). Lo que nos interesa calcular es la probabilidad $P(LCL < S^2 < UCL|\sigma^2 = \sigma_1^2)$ y como la variable aleatoria definida como $\frac{(n-1)S^2}{\sigma^2}$ tiene distirbución χ^2 con $n-1$ grados de libertad podemos calcular la probai¿bilidad así:

$$P\left(LCL < \frac{(n-1)S^2}{\sigma^2} < UCL|\sigma^2 = \sigma_1^2\right) = P\left(LCL < \frac{(n-1)S^2}{\sigma_1^2} < UCL\right)$$

$$= P\left(\frac{(n-1)LCL}{\sigma_1^2} < S^2 < \frac{(n-1)UCL}{\sigma_1^2}\right)$$

$$= P\left(\chi^2_{(n-1)} < \frac{(n-1)UCL}{\sigma_1^2}\right) - P\left(\chi^2_{(n-1)} < \frac{(n-1)LCL}{\sigma_1^2}\right)$$

Y calculando estas probabilidades podremos graficar las curvas OC, con el fin de realizar esta operación de forma más sencilla se crea la siguiente función en R:

```
OCs2 = function(n = 5, sigma2_0 = 1, corrimiento = 1){
  UCL = sigma2_0/(n-1) * qchisq(0.975, df = n - 1)
  LCL = sigma2_0/(n-1) * qchisq(0.025, df = n - 1)
  beta = pchisq((n-1) * UCL / corrimiento , df = n - 1)
  - pchisq((n-1) * LCL / corrimiento , df = n - 1)
  return(beta)
}
```

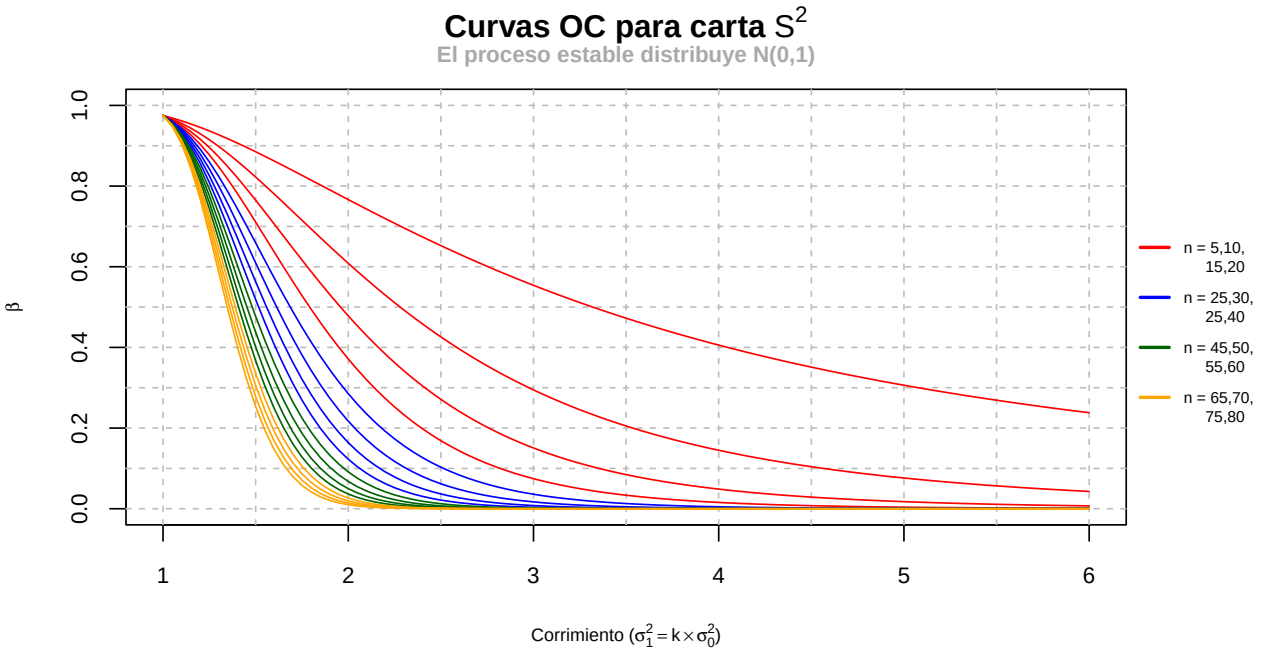

Figura 3: Curvas de operación característica para carta S²

Note que el incremento en el tamaño de los subgrupos racionales se realizo en múltiplos de 5 a cambio de obtener un gráfico más entendible y útil. Respecto al gráfico como tal, vemos que la carta

S^2 es más efectiva para muestras lo suficientemente grandes y además de esto, vemos también que a partir de tamaños de alrededor de 50 no es realmente productivo intentar aumentar n pues la relación costo-beneficio podría no ser óptima. Finalmente, revisando el gráfico nos damos cuenta que para corrimientos $\sigma_1^2 = 1{,}5\ \sigma_0^2$ se obtiene una probabilidad de detectar el cambio en el siguiente subgrupo racional de más del 50 % para $n > 15$ por lo que sería recomensable usar esta carta para procesos en que se permitan tomar más de 15 muestras.

Ejercicio 3: Sea $X \sim N(\mu_0, \sigma_0)$ una característica de calidad. Construya la carta $\bar{X}$ para el monitoreo de la media del proceso. Genere 10 muestras de tamaño n provenientes de X, de tal modo que la media muestral de ninguna de ellas caiga fuera de los límites de control. A partir del undécimo momento de monitoreo se pide generar muestras del mismo tamaño n provenientes de una distribución normal con media $\mu_1 = \mu_0 + k\sigma_0$ y $\sigma_1 = \sigma_0$ (con $k = 1, 0$) hasta que la carta emita una señal por primera vez. Si se asume que el proceso caracterizado por X es estable y que se desconoce el momento en el cual se produjo el incremento en el nivel medio, ¿en qué muestra ocurrió el cambio en la media del proceso más probablemente?

Tal como se pide en el ejercicio, se plantea el siguiente algoritmo en R para k = 1.

```r
# Parámetros
mu_0 <- 0; sigma_0 <- 1; k <- 1; n <- 5
LCL <- mu_0 - 3 * (sigma_0 / sqrt(n)); UCL <- mu_0 + 3 * (sigma_0 / sqrt(n))

generar_muestra <- function(mu, sigma, size) {
  return(rnorm(size, mean = mu, sd = sigma))
}

muestras <- list()   # Primeras 10 muestras
for (i in 1:10) {
  muestra <- generar_muestra(mu_0, sigma_0, n)
  media_muestra <- mean(muestra)
  muestras[[i]] <- muestra
}

muestra_num <- 11; set.seed(13) #  muestra num11 en adelante con cambio en la media del proceso
while (TRUE) {
  muestra <- generar_muestra(mu_0 + k * sigma_0, sigma_0, n)
  media_muestra <- mean(muestra); muestras[[muestra_num]] <- muestra
  if (media_muestra < LCL || media_muestra > UCL) {
    cat("Señal fuera de los límites de control en la muestra", muestra_num, "\n")
    break
  }
  muestra_num <- muestra_num + 1
}
```

```
## Señal fuera de los límites de control en la muestra 11
```

```
## La carta dio señal en el tiempo 11
```

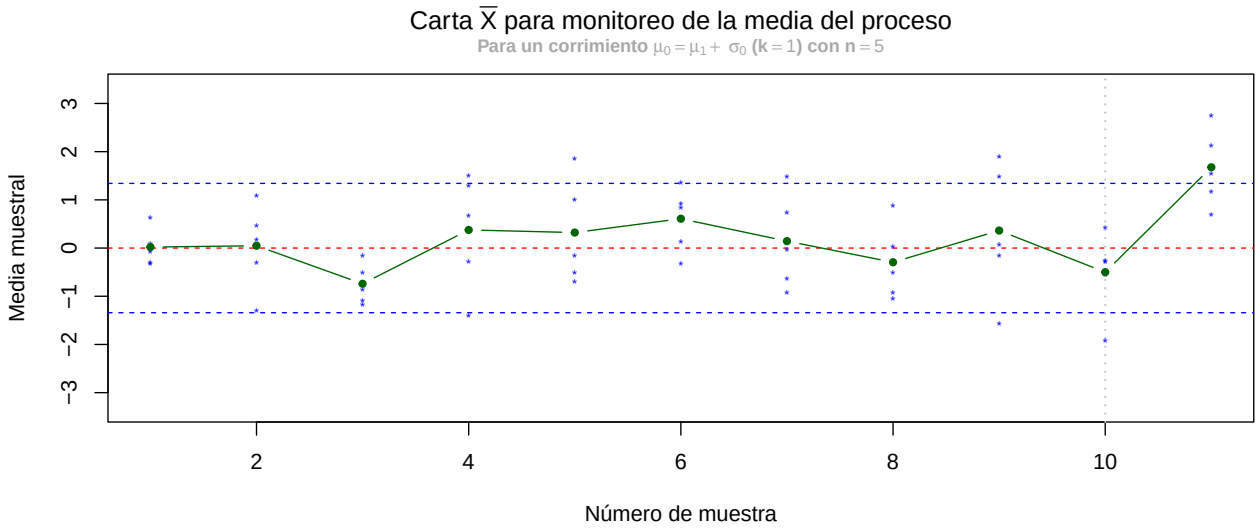

Figura 4: Carta X barra (Con k = 1)

Y para $k = 2$ simplemente se cambia este valor en los parámetros establecidos al inicio del algoritmo. Obteniendo la siguiente gráfica

```
## Señal fuera de los límites de control en la muestra 45

## La carta dio señal en el tiempo 45
```

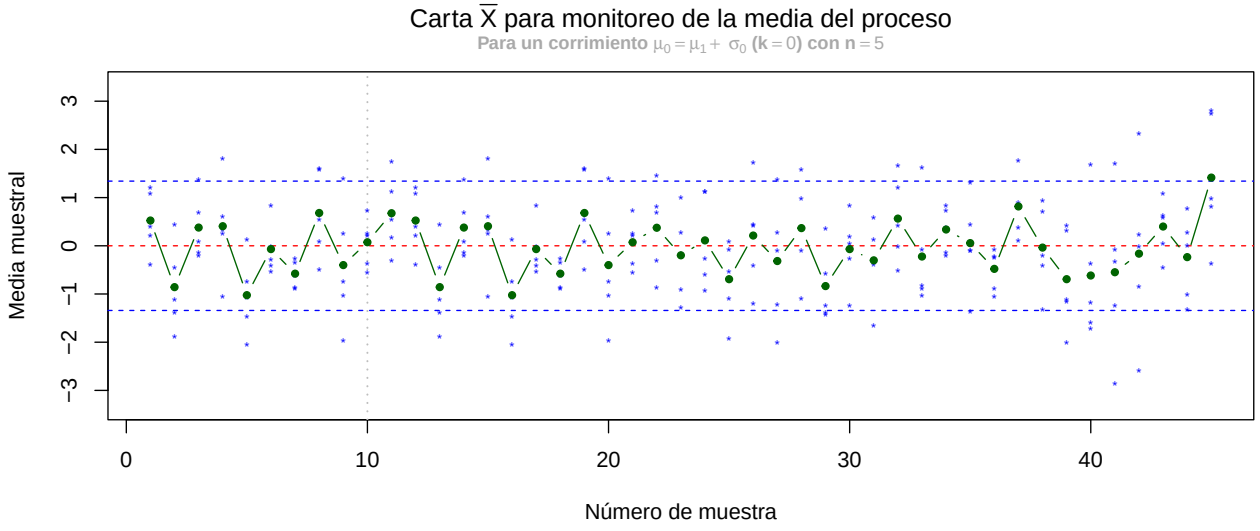

Figura 5: Carta X barra (Con k = 0))

Finalmente el momento en que más probablemente se dio el cambio en la media resultará del estimador $\hat{\tau}_{MV} = \underset{0 \leq t < T}{\operatorname{argmax}} \left[(T-t)(\bar{\bar{X}}_{T,t} - \mu_0)^2 \right]$ implementado en R así:

```r
# Para k = 1
Total = length(medias)
FV = c()
for (i in 0:(Total - 1)){
  media_no_control = sum(medias[(i+1):Total])/(Total - i)
  FV = c(FV,log((Total-i) * (media_no_control - mu_0)^2))
}
```

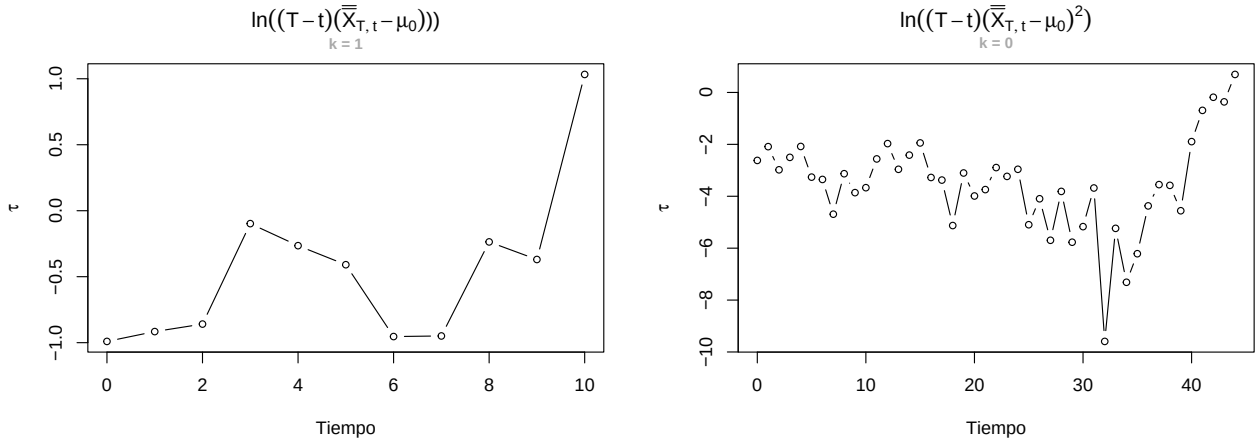

Figura 6: Gráfica función de verosimilitud para punto de cambio (k = 1,0)

Vemos que la función para valores de k pequeños tiende a ser maximizada más atrás de la última media registrada en la carta, pero para valores de k más grandes tiende a maximizarse en los últimos valores de la carta.

Ejercicio 4: Sea $X \sim N(\mu_0, \sigma_0)$ una característica de calidad. Se pide:

a. Mediante simulaciones, establezca el comportamiento del ARL de la Carta $\bar{X}$ con límites tres sigma para observaciones normales.

b. Genere 20 subgrupos racionales de tamaño $n = 3$ provenientes de X. Asúmase que el proceso es estable en cuanto a dispersión y con los subgrupos iniciales, construya la carta $\bar{X}$ como es habitual hasta verificar la estabilidad del proceso. Establezca el comportamiento del ARL para la carta que se obtiene del análisis de Fase I realizado.

c. Repetir lo indicado en el literal (b) con 50 subgrupos racionales de tamaño $n = 3$. Comente los resultados.

En cuanto al punto a:

Se nos dice que el ARL_0 es una medida igual a $\frac{1}{P(\bar{X}>UCL \text{ o } \bar{X}<LCL|\mu=\mu_0)}$, y esta medida nos da un estimado de en qué número de pruebas se hallará la primera señal de alarma. Para encontrar esta probabilidad, tomamos una muestra particularmente grande de una población normal estándar, con subgrupos grandes, y contamos cuántas de estas medias en los subgrupos salen de los límites pedidos, con límites tres sigma.

```r
library(Matrix)
mu_0 <- 0; sigma_0 <- 1  # Estableciendo parámetros verdaderos

# Parámetros de la simulación
n_samples <- 10000; sample_size <- 1000; control_limit <- 3

k <- seq(0, 0.18, by = 0.02)
UCL <- mu_0 + control_limit * (sigma_0/sqrt(sample_size))
LCL <- mu_0 - control_limit * (sigma_0/sqrt(sample_size))

means <- numeric(n_samples); all <- numeric(n_samples * sample_size)
for (i in 1:n_samples) {
  samples <- rnorm(sample_size, mean = mu_0, sd = sigma_0)
  means[i] <- mean(samples)
  all[((i - 1) * sample_size + 1):(i * sample_size)] <- samples
}
counts <- matrix(0, nrow = n_samples, ncol = length(k))
for (j in 1:length(k)){
  for (i in 1:n_samples) {
```

```
    counts[i , j] <- as.integer(means[i] + k[j] > UCL | means[i] + k[j] < LCL)
  }}

Pf <- colMeans(counts)
ARL_0 <- 1/Pf[1]; ARL_1 <- 1/(1-Pf)
```

Para los ARL_0 nos da la salida con los diferentes Valores para cada k diferente y una grafica donde vemos el aumento del ARL con forme el valor de k se incrementa.

```
Pf <- colMeans(counts);Pf
[1] 0.0025 0.0118 0.0439 0.1369 0.3249 0.5646 0.7922 0.9231
ARL_1 <- 1/(1-Pf); ARL_1
[1]  1.002506  1.011941  1.045916  1.158614  1.481262  2.296739  4.812320 13.003901
```

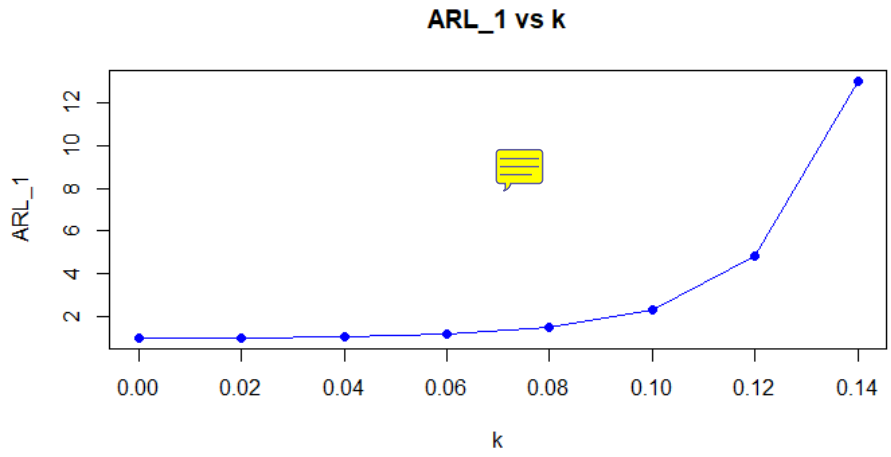

Figura 7: Pruebnorm2

Podemos ver que el ARL para una muestra no controlada mientras mas fuera de control esta (Ergo que su valor k sea mas alto) Su ARL_1 Aumentara en mayor proporcion.

Al terminar con este codigo no da la salida: ARL_0 [1]370,370 lo que nos señala que en estimado para una distribucion normal estandar lo probable es que una alrededor de el numero de media 370 se sale de control.

Para comprobarlo hacemos una simulacion de datos de esta distribucion hacemos que pare cuando una media sale de control y ver la media en cuanto a las medias necesarias para que se saliera de control

```
resultados <- numeric(length(k))
for (z in 1:length(k)) {
  i <- 1; jj <- numeric(n_samples)
  while (i <= n_samples) {
    j <- 1
    while (j <= n_samples) {
      Prueb <- rnorm(sample_size, mean = mu_0, sd = sigma_0)
      if (mean(Prueb) + k[z] < LCL | mean(Prueb) + k[z] > UCL) {
        break  # Detiene el bucle
      }
      j <- j + 1
    }
    jj[i] <- j; i <- i + 1
  }
  indices <- which(jj != 0)
```

```
  jj <- jj[indices]
  resultados[z] <- mean(jj)
}
```

Este codigo nos da que de tantas muestras en promedio fallaron en su media numero 370 lo que comprueba el ARL_0 estimado con simulaciones.

Tambien podemos ver como la media con del valor en el que manda alerta la carta es mas baja mientras mayor sea el valor de K comprobando nuestros valores para ARL_1. Tal como lo vemos en la gráfica:

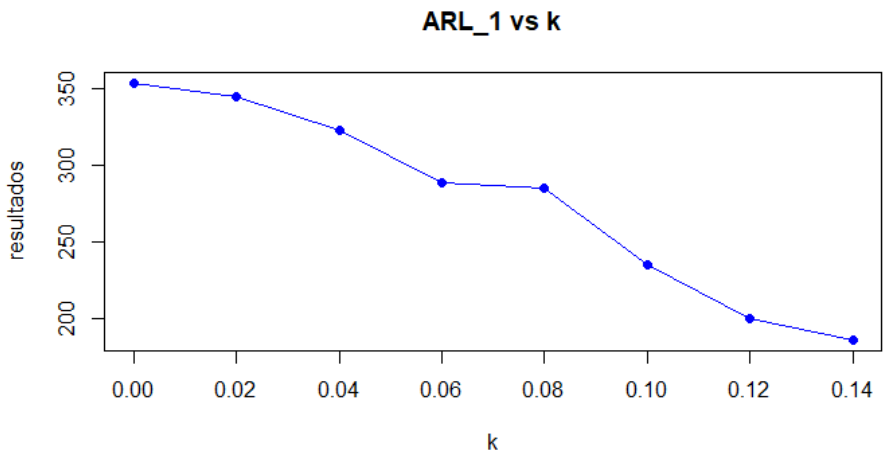

Figura 8: Medias de fallo

Respecto al punto b usando las especificaciones pedidas en el enunciado, repetimos exactamente el mismo proceso usado en el punto anterior solo cambiando los valores de el tamaño de muestra y el numero de muestra.

Nuestra primera evidencia es los resultados que nos da el codigo.

```
Pf <- colMeans(counts);Pf
[1] 0.00 0.05 0.10 0.10 0.30 0.45 0.60 0.75
ARL_0 <- 1/Pf[1];ARL_0
[1] Inf
ARL_1 <- 1/(1-Pf)
1.000000 1.052632 1.111111 1.111111 1.428571 1.818182 2.500000 4.000000
```

Podemos ver que al igual que el punto anterior su ARL_1 esta aumentando miestras mayor sea su valor k y la media del valor en que manda alerta tambien se va reduciendo mientras mayor sea el valor de k, las principales diferencias son dos

a. Tuvimos que vernos en la obligacion de aumentar los niveles de k dado que si usabamos los mismos que en el punto anterior estos nos darian en su mayoria valores muy similares y bajos cercanos a cero.
b. Su valor para ARL_0 es indefinido esto debido a que 20 muestras no son suficientes para estimar la probabilidad de alerta mediante simulaciones.

Podemos ver que sigue las mismas tendencias pero sus valores se ven influenciados por lo pequeño de su numero de muestras.

Respecto al literal c:

Al igual que en los dos antiguos enunciados se uso el mismo codigo solo diferenciado por su tamaño y numero de muestras. vemos con sus resultado y graficas que se mantiene las tendencias con respecto

al cambio de k, pero sus resultados son mas cercanos que los del punto b. Aunque al igual que con dicho punto este sigue sin poder aproximar un valor finito para ARL_0.

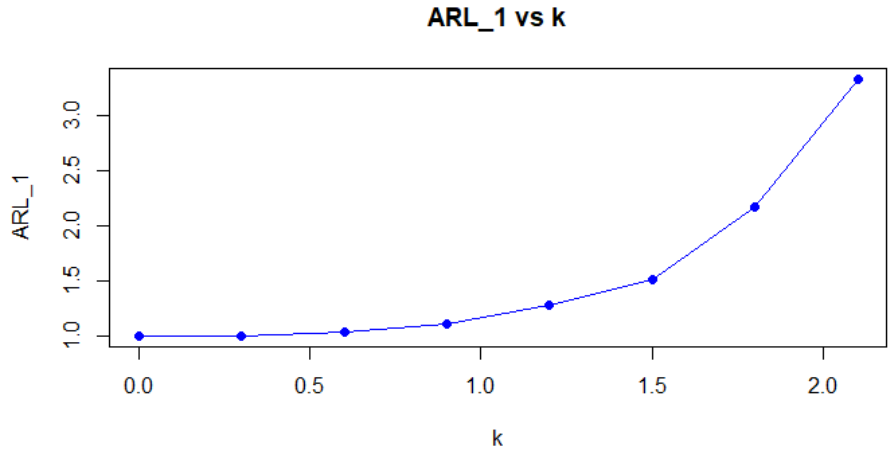

Figura 9: ARL_1 para $n = 50$

Podemos ver que el comportamiento del ARL se mantiene para todos lo numero de muestras pero varia su acercamiento a el valor estimado.

Ejercicio 5: Calcular el ARL de la Carta $\bar{X}$ mediante cadenas de Markov. Diseñar la carta con límites de control ubicados a tres desviaciones estándar de la media y dividiendo la región de control estadístico en franjas de ancho igual a una desviación estándar.

Creamos la cadena de Markov utilizando las medias de muestra en la simulacion para el punto 4 hasta su primer punto donde la carta suelta una alerta. Esto para tener en la matriz de transicion el estado absorbente fijo.

```
mu_1 <- mean(means); sds <- sd(means)
UCL2 <- mu_1 + 3 * sds; LCL2 <- mu_1 - 3 * sds  # Límites de control
franja_ancho <- sds; franja_sup <- seq(LCL2, UCL2, by = franja_ancho)  # Franjas de control

vector <- 1:n_samples; datos2 <- data.frame(vector,means)
x.lim <- c(0,n_samples); y.lim <- c(franja_sup[1]-0.01,franja_sup[7]+0.01)

estados1 <- rep(0, length(means)); i <- 1
while (i <= length(means)) {
  if (means[i] < franja_sup[1] | means[i] > franja_sup[7]) {
    estados1[i] <- 4
    break  # Detiene el bucle
  }
  if (means[i] > franja_sup[2] & means[i] < franja_sup[6]) { estados1[i] <- 1 }
  if (means[i] > franja_sup[6] & means[i] < franja_sup[7]) { estados1[i] <- 2 }
  if (means[i] > franja_sup[1] & means[i] < franja_sup[2]) { estados1[i] <- 3 }
  i <- i + 1
}
indices_no_cero <- which(estados1 != 0); estados1 <- estados1[indices_no_cero]
observed_table <- table(estados1[-length(estados1)], estados1[-1])
#Matriz de transcision
T_ <- observed_table / rowSums(observed_table)

T_0 <- matrix(0, nrow = 4, ncol = 4, byrow = FALSE)
T_0[1:nrow(T_), 1:ncol(T_)] <- T_; T_0[4,4]<- 1

R <- T_0[1:3, 1:3] ; I <- diag(1,nrow(R)); t1 <- rep(1, times = 3)
```

```
R1 <- solve(I - R); tP1 <- T_0[1, 1:3]; tP2 <- T_0[2, 1:3]; tP3 <- T_0[3, 1:3]

ARL1 <- tP1 %*% R1 %*% t1
ARL2 <- tP2 %*% R1 %*% t1
ARL3 <- tP3 %*% R1 %*% t1; ARL1; ARL2; ARL3
```

Aspectos que evidenciamos con la fabricacion de este codigo.

a. Como se puede ver en el codigo no usamos 5 areas de estado como se veria dividiendo las franjas en una desviacion de ancho, esto debido a la necesidad que en el estado 1 y 2, de las areas de riesgo su dato inmediatoa anterior no se encontraran en su mismo estado, algo que solo se cumplia tomando en cuentas dicgos dos rangos.
b. Otro aspecto importante es que si no existe un dato salido de control y dando alerta, la matriz inversa necesaria para sacar el ARL no existe.
c. El punto anterior se liga al hecho que este codigo al hacer la operacion matricial para encontrar el ARL dado en clase no nos da una aproximacion de cuando podria dar un valor que lance la alerta si no el dato exacto donde Si que lo hizo.

Tomemos como ejemplo el resultado que nos da el codigo para una muestra ARL1 = ARL2 = ARL3 = 55 (Uno para cada estado), y podemos ver la veracidad de esto con un grafico simple.

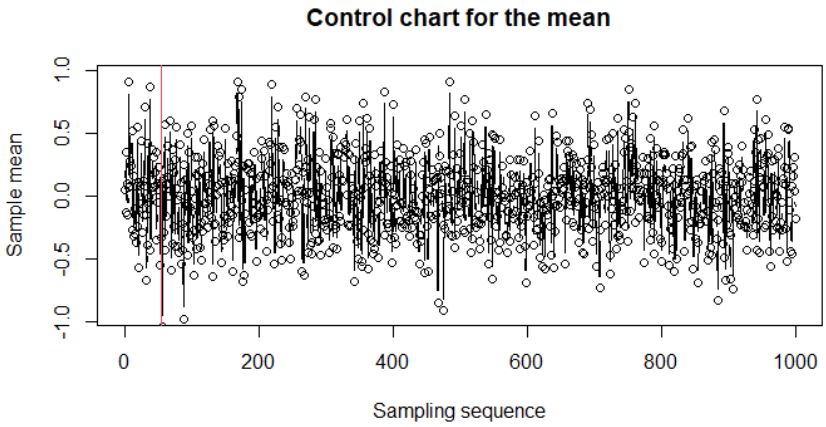

Figura 10: Ejemplo markov

En conclusión este metodo es bueno para saber con exactitud donde dio la alerta un proceso mas no para estimarlo.