Examen Parcial de Estadística Computacional

Jaime Gomez Marin

# Pregunta 2 :

1. Haga un programa que simule el experimento y muestreo 10 repeticiones

# Primera cajaPilaA  
p1 <- 20/24  
q1 <- 4/24  
  
# Segunda cajaPilaB  
p2 <- 17/20  
q2 <- 3/20  
  
# Tercera cajaPilaC  
p3 <- 17/22  
q3 <- 5/22  
  
  
cajaPilaA <- c(rep("pila\_buena",20), rep("pila\_mala",4))  
cajaPilaB <- c(rep("pila\_buena",17), rep("pila\_mala",3))  
cajaPilaC <- c(rep("pila\_buena",17), rep("pila\_mala",5))  
  
experimentoCajaD <- function(cajaPilaA, cajaPilaB, cajaPilaC){  
   
 m\_cajaPilaA <- sample(cajaPilaA,4)  
  
 m\_cajaPilaB <- sample(cajaPilaB,4)  
  
 m\_cajaPilaC <- sample(cajaPilaC,4)  
  
   
 cajaPilaD = cbind(m\_cajaPilaA,m\_cajaPilaB,m\_cajaPilaC)   
   
 return(table(cajaPilaD))  
   
}  
  
# 10 repeticiones  
for ( i in 1:10) {   
 print(i)  
 print(experimentoCajaD(cajaPilaA, cajaPilaB, cajaPilaC))  
}

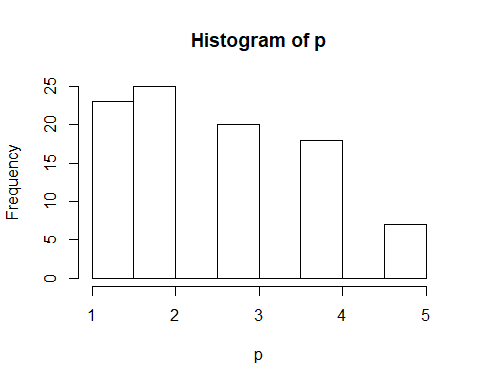
## [1] 1  
## cajaPilaD  
## pila\_buena pila\_mala   
## 9 3   
## [1] 2  
## cajaPilaD  
## pila\_buena pila\_mala   
## 10 2   
## [1] 3  
## cajaPilaD  
## pila\_buena pila\_mala   
## 11 1   
## [1] 4  
## cajaPilaD  
## pila\_buena pila\_mala   
## 10 2   
## [1] 5  
## cajaPilaD  
## pila\_buena pila\_mala   
## 7 5   
## [1] 6  
## cajaPilaD  
## pila\_buena pila\_mala   
## 10 2   
## [1] 7  
## cajaPilaD  
## pila\_buena pila\_mala   
## 10 2   
## [1] 8  
## cajaPilaD  
## pila\_buena pila\_mala   
## 11 1   
## [1] 9  
## cajaPilaD  
## pila\_buena pila\_mala   
## 11 1   
## [1] 10  
## cajaPilaD  
## pila\_buena pila\_mala   
## 11 1

1. Estime la distribución de probabilidad de X

N <- 100  
  
p <- c(rep(0,N))  
  
# N repeticiones  
for ( i in 1:N) {   
 res = experimentoCajaD(cajaPilaA, cajaPilaB, cajaPilaC)  
 p[i] = res["pila\_mala"]  
}  
  
print(p)

## [1] 4 1 3 2 4 1 3 2 NA 4 NA 3 2 4 1 2 4 3 4 2 3 1 1  
## [24] 4 2 2 3 NA 3 1 4 2 3 5 NA 2 3 4 3 1 4 4 4 5 NA 2  
## [47] 1 5 2 3 1 2 2 5 1 2 2 2 3 3 NA 1 1 3 2 1 1 4 3  
## [70] 1 2 5 2 5 1 1 2 3 4 1 4 5 1 1 1 3 3 3 2 1 3 2  
## [93] 1 4 4 2 NA 2 4 2

hist(p)

 Se puede observar que la probabilidad de encontrar pilas defectuosas en la caja D siguen una distribución exponencial

# Pregunta 3 :

1. Haga un programa que calcule el valor del estimador C:

datos = c(13.30, 13.10, 12.95, 13.45,  
 13.35, 13.15, 13.10, 13.50,  
 13.40, 13.20, 13.05, 13.55,  
 12.75, 13.25, 12.60, 13.60,  
 12.80, 12.84, 12.65, 13.65,   
 12.85, 12.78, 12.70, 13.70,  
 12.90, 13.00, 12.90, 13.75)   
  
  
  
# Funcion para calcular el estimador  
estimadorC <- function(datos) {  
  
 n <- length(datos)  
 mu <- mean(datos)  
   
 a <- n\*sum((datos-mu)^3)^2  
 b <- sum((datos-mu)^2)^3  
   
 ret <- sqrt(a/b + 0.5)  
  
 return (ret)  
}  
  
estimadorC(datos)

## [1] 0.7443395

1. Estime el intervalo de confianza para el coeficiente C al 98% de confianza a traves de los métodos estandar , estudentizado y doblemenete estudentizado

# Estimador bootstrap para el estimador C  
calc\_bootstrap <- function(datos,B) {  
  
 n = length(datos)  
 estimac=rep(0,B)  
   
 for (i in 1:B){  
 muestra = sample(datos,n,T)  
 estimac[i] = estimadorC(muestra)  
 }  
   
 estboot=mean(estimac)  
 eeboot=sd(estimac)  
  
 return(list(estimador=estboot,eestandar=eeboot, vector=estimac))  
  
}

# Método estandar

calc\_ic\_met\_estandar <-function(datos, B, alpha ) {  
   
 bstrap = calc\_bootstrap(datos,B)  
   
 media = bstrap$estimador  
 desvStd = bstrap$eestandar  
   
 lis = media - qnorm(1-alpha/2)\*desvStd  
 lss = media + qnorm(1-alpha/2)\*desvStd  
 rango = lss - lis  
   
 return(data.frame("metodo"="normal", lis, lss, rango))  
}

# Método estudentizado

calc\_ic\_met\_estudentizado <-function(datos, B, alpha ) {  
  
 bstrap = calc\_bootstrap(datos,B)  
   
 estCstd = bstrap$eestandar  
 estCvector = bstrap$vector  
 estC = estimadorC(estCvector)  
   
 testud=rep(0,B) # Inicializa el estudentizado  
   
 testud = (estCvector-estC)/estCstd  
   
 lis = estC + quantile(testud,alpha/2)\*estCstd   
 lss = estC + quantile(testud,1-alpha/2)\*estCstd   
   
 rango = lss - lis  
   
 return(data.frame("metodo"="estudentizado", lis, lss, rango))  
}

# Método doblemente estudentizado

# Calcular IC usando el metodo de doble estudentizado  
calc\_ic\_met\_doble\_estudentizado <- function(datos,B1,B2,alpha) {  
  
 n = length(datos)  
 estimacB1 = rep(0,B1)  
 errorB1 = rep(0,B1)  
   
 for (i in 1:B1){  
 muestraB1 = sample(datos,n,T)  
 estimacB1[i] = estimadorC(muestraB1)  
   
 estimacB2 = rep(0,B2)  
 for (k in 1:B2) {  
 muestraB2 = sample(muestraB1,n,T)   
 estimacB2[k] = estimadorC(muestraB2)  
 }  
   
 errorB1[i] = sd(estimacB2)  
 }  
  
 testud = rep(0,B1) # t estudentizado , se hace para cada unos de la p muestras boostrap  
 estimacBASE = estimadorC(datos)  
   
 testud = (estimacB1-estimacBASE)/errorB1  
   
 estCstd = sd(estimacB1)  
  
 lis = estimacBASE + quantile(testud,alpha/2)\*estCstd   
 lss = estimacBASE + quantile(testud,1-alpha/2)\*estCstd   
   
 rango = lss - lis  
   
 return(data.frame("metodo"="doble\_estudentizado", lis, lss, rango))  
}

set.seed(1000)  
  
alpha <- 1-0.98  
B1 <- 100   
B2 <- 50  
  
df = rbind(calc\_ic\_met\_estandar(datos,B1,alpha),  
 calc\_ic\_met\_estudentizado(datos,B1,alpha),  
 calc\_ic\_met\_doble\_estudentizado(datos,B1,B2,alpha))  
  
df

## metodo lis lss rango  
## 1 normal 0.5422957 1.0540464 0.5117507  
## 1% estudentizado 0.7071101 1.1285076 0.4213975  
## 1%1 doble\_estudentizado 0.6655138 0.9985352 0.3330214

Se puede apreciar que el intervalo de confianza del metodo doblemente estudentizado tiene menor variabilidad.