

Ejercicio 7.1

Augusto Souto

24 de abril de 2019

Índice

1. Descripción General del Problema	2
2. Parte A	2
2.1. Pseudocódigo	2
2.2. Código-Programa	4
3. Parte B	5
3.1. Introducción de Datos	5
3.2. Estimaciones	5

1. Descripción General del Problema

Problema: para diseñar un Sistema Nacional de Areas Protegidas, uno de los modelos empleados tiene en cuenta por un lado un conjunto $E = 1, \dots, e$ de especies que se desea preservar, y por otra parte un conjunto Z de zonas donde es posible establecer una reserva. La relación entre ambos conjuntos está dada por una matriz $P = ((p_{ij}))$, con i en Z y j en E , tal que $z_{ij} = 1$ es en la zona i se ha observado la presencia de la especie j .

Es interesante elegir un conjunto de M zonas, tales que todas las especies estén representadas en al menos una zona. La determinación del menor M que hace posible esta propiedad es un problema de “set covering” (NP-difícil), que escapa el alcance de este curso.

Suponiendo que el valor de M ya fue elegido, un segundo nivel es ver cuantas formas distintas hay de elegir este conjunto de zonas.

Para esto, es posible aplicar el método Monte Carlo para, dado el cardinal de E , el cardinal de Z , la matriz $P = ((p_{ij}))$, y un valor de M predeterminado, estimar cuántas combinaciones de M zonas distintas cumplen la propiedad requerida (representar todas las especies).

Se debe recibir en entrada el número de replicaciones a realizar, y el nivel de confianza; en salida, se debe dar la estimación del número de combinaciones $NC(M)$, así como la desviación estándar y un intervalo de confianza (del nivel especificado) calculado en base al criterio de Agresti-Coull.

2. Parte A

Escribir un programa para hacer el cálculo previamente descrito. Entregar pseudocódigo y código.

Comentario: para el muestreo uniforme de los subconjuntos de Z , se debe hacer un “muestreo sin remplazos” de las distintas zonas. Es posible hacerlo sorteando siempre números uniformes entre 1 y z , y descartando aquellos ya elegidos, aunque se pierde eficiencia por los sorteos repetidos. Otra forma más eficiente es elegir primero de forma uniforme un número de 1 a z , y eliminar esta zona de Z ; el segundo sorteo hacerlo entre 1 y z menos 1; el tercero entre 1 y z menos 2, y así sucesivamente, donde en cada etapa solo se elige entre las zonas que todavía no fueron sorteadas.

2.1. Pseudocodigo

```
##PSEUDOCODIGO##

Define matrix M with a rows and b columns
Set n
set m
set confianza

define function estimacion(Matrix, n,m, confianza){

  r<-combinations of a taken m

  define matrix sumas with 1 row and b columns

  define matrix Muestra with m rows and b columns

  Set S=0

  for (i in 1:n) {
    set seed
    sample m rows from M and fill matrix Muestra with M[sample,]
```

```

for (j in 1:ncol(muestra)) {
  sumas[,j]=sum(Muestra[,j])
}

if(sumas[1,j]>0 for all j) then S=S+1 else S=S
}

valor_est<-r*S/N
var_est<-valor_est*(r-valor_est)/(n-1)

z2<-(inverse normal at quantile 1-confianza/2)^2

ene<-z2+n

pe<-(1/ene)*(S+(z2/2))

sup<-pe+sqrt(z2)*(pe/ene*(1-pe))^0.5
inf<-pe-sqrt(z2)*(pe/ene*(1-pe))^0.5

sup<-sup*r
inf<-inf*r

#DEFINICION DE TODAS LAS VARIABLES DE SALIDA:

#VALOR ESTIMADO
#VARIANZA ESTIMADA
# LIMITES DEL INTERVALO DE CONFIANZA DE AGRESTI COULL AL 95%

return(c(valor_est, var_est, sup, inf))
}
##

```

2.2. Codigo-Programa

```
##EJERCICIO 1-CODIGO#
rm(list=ls())

##parametros de entrada##

#Matriz M
#n corridas
#m zonas
#confianza

estimar<-function(M, n, m, confianza){

  set.seed(1)

  r<-factorial(nrow(M))/(factorial(nrow(M)-m)*factorial(m)) #dimension#

  sumas<-matrix(0,nrow=1,ncol=dim(M)[2])

  muestra<-matrix(nrow = m, ncol=dim(M)[2])

  S=0 #acumulador#

  for (i in 1:n) {
    for (j in 1:dim(M)[2]) {

      muestra= M[sample(nrow(M), size = m, replace = F),]

      sumas[,j]= sum(muestra[,j])

    }

    if(all(sumas>0)){S=S+1}else{S=S} #check de la condicion, todos los animales deben
                                     #estar contenido en el set de zonas.
  }

  valor_est<-S*r/n
  var_est<-valor_est*(r-valor_est)/(n-1)

  z2<-qnorm(1-confianza/2)^2
  ene<-z2+n

  pe<-(1/ene)*(S+(z2/2))

  sup<-pe+qnorm(1-confianza/2)*(pe/ene*(1-pe))^0.5
  inf<-pe-qnorm(1-confianza/2)*(pe/ene*(1-pe))^0.5

  sup<-sup*r
  inf<-inf*r

  return(c(valor_est, var_est, sup, inf))
}
```

3. Parte B

Sea el siguiente caso: $z = 15$ zonas, $e = 8$ especies, $p_{ij} = 1$ para los nodos (ver código).

Usando el programa anterior, y empleando 1000 replicaciones de Monte Carlo, estimar los valores de NC(M) para $M = 5$ y para $M = 6$ con intervalos de confianza de nivel 95 %.

3.1. Introducción de Datos

```
#introduzco la matriz
matriz=matrix(0,nrow = 15, ncol = 8)
matriz[1,1]=matriz[1,2]=matriz[1,3]=matriz[1,6]=matriz[1,8]=1
matriz[2,1]=matriz[2,3]=matriz[2,4]=matriz[2,6]=matriz[2,7]=1
matriz[3,1]=matriz[3,4]=matriz[3,5]=matriz[3,7]=matriz[3,8]=1
matriz[4,1]=matriz[4,2]=matriz[4,4]=matriz[4,6]=matriz[4,7]=matriz[4,8]=1
matriz[5,1]=matriz[5,3]=matriz[5,6]=matriz[5,8]=1
matriz[6,2]=matriz[6,4]=matriz[6,7]=matriz[6,8]=1
matriz[7,3]=matriz[7,5]=matriz[7,8]=1
matriz[8,2]=matriz[8,3]=matriz[8,5]=matriz[8,7]=matriz[8,8]=1
matriz[9,2]=matriz[9,5]=matriz[9,6]=matriz[9,8]=1
matriz[10,3]=matriz[10,4]=matriz[10,6]=matriz[10,7]=matriz[10,8]=1
matriz[11,2]=matriz[11,5]=matriz[11,6]=matriz[11,7]=1
matriz[12,3]=matriz[12,4]=matriz[12,5]=matriz[12,6]=matriz[12,7]=1
matriz[13,1]=matriz[13,2]=matriz[13,6]=matriz[13,7]=matriz[13,8]=1
matriz[14,1]=matriz[14,2]=matriz[14,4]=matriz[14,6]=matriz[14,7]=1
matriz[15,2]=matriz[15,3]=matriz[15,5]=matriz[15,6]=matriz[15,7]=matriz[15,8]=1
```

3.2. Estimaciones

Una vez introducida la matriz, podemos usarla junto a los otros argumentos de la función para estimar el número de combinaciones, su varianza y su intervalo.

```
#cargo dos paquetes, kableextra para
#poner los resultados en una tabla y magrittr para usar el operador "%>%"

if (!require('kableExtra')) install.packages('kableExtra')
if (!require('magrittr')) install.packages('magrittr')

valores_5<-as.matrix(estimar(M=matriz, n=1000, m=5, confianza=0.05))

valores_6<-as.matrix(estimar(M=matriz, n=1000, m=6, confianza=0.05))
tabla<-cbind(valores_5, valores_6)

rownames(tabla)<-c("Estimacion", "Varianza", "lim superior", "lim inferior" )
colnames(tabla)<-c("5" , "6")

kableExtra::kable(tabla, caption = "Estimaciones Finales") %>%
  kable_styling(position = "center", latex_options = "hold_position", font_size = 15)
```

Los resultados se presentan en la siguiente tabla:

Cuadro 1: Estimaciones Finales

	5	6
Estimacion	2834.8320	4899.895
Varianza	477.2052	515.519
lim superior	2873.0458	4936.968
lim inferior	2786.4135	4844.474