Estimación por Sistemas Múltiples

Paulo Sergio García Méndez

Octubre de 2024

1 Modelos Loglineales

1.1 Introducción

La parte teórica de este documento se basa en notas del curso: https://newonlinecourses.science.psu.edu/stat504/node/49/. El objetivo es proporcionar los fundamentos de los modelos estadísticos loglineales y, después, mostrar una aplicación para estimar el tamaño de una población de difícil alcance como lo es la de las víctimas de trata de personas. Esta técnica se conoce como Estimación por Sistemas Múltiples (Multiple System Estimation, en inglés). Los ejemplos incluidos están implementados en el lenguaje R y se incluye el código para replicarlos.

1.2 Modelos Loglineales para dos variables

Los modelos loglineales para dos variables describen las asociaciones y los patrones de interacción entre dos variables aleatorias categóricas. Sea μ_{ij} el conteo esperado, $E(n_{ij})$ en una tabla $I \times J$ de dos variables aleatorias A y B.

Objetivo: Modelar los conteos por celdas $\mu_{ij} = n\pi_{ij}$.

Estructura del modelo: El modelo loglineal saturado para dos variables con interacción es

$$\log(\mu_{ij}) = \mu + \alpha_i + \beta_j + \gamma_{ij}$$

o en notación introducida por Agresti:

$$\log(\mu_{ij}) = \lambda + \lambda_i^A + \lambda_j^B + \lambda_{ij}^{AB}$$

donde $i = 1, ..., I, j = 1, ..., J$, son niveles de las variables aleatorias categóricas A y B, con restricciones

$$\sum_i \lambda_i = \sum_j \lambda_j = \sum_i \sum_j \lambda_{ij} = 0$$

para controlar la sobreparametrización. Esto último significa que el número de parámetros es mayor a los que se pueden estimar de manera única.

1.2.1 Supuestos del modelo

Las $N = I \times J$ conteos en las celdas son observaciones independientes de una variable aleatoria Poisson $n_{ij} \sim Poisson(\mu_{ij})$.

1.2.2 Modelos Loglineales para dos variables

Dadas dos variables aleatorias categóricas A y B, existen dos tipos de modelos que consideraremos:

- Modelo de independencia (A, B)
- Modelo saturado (AB)

1.3 Modelo de Independencia para tablas de dos vías

Recordemos que la independencia se puede enunciar en términos de las probabilidades de las celdas como un producto de probabilidades marginales,

$$\pi_{ij} = \pi_{i+}\pi_{+j} \quad i = 1, ..., I, j = 1, ..., J$$

y en términos de las frecuencias de las celdas,

$$\mu_{ij} = n\pi_{ij} = n\pi_{i+}\pi_{+j} \quad i = 1, ..., I, j = 1, ..., J$$

.

Al aplicar logaritmo natural en ambos lados de la igualdad, obtenemos el **modelo loglineal de independencia**:

$$\log(\mu_{ij}) = \log(n) + \log(\pi_{i+}) + \log(\pi_{+j})$$
$$\log(\mu_{ij}) = \lambda + \lambda_i^A + \lambda_j^B$$

donde los superíndices A y B solamente denotan las dos variables categóricas (Agresti). Este modelo implica que todas las razones de momios deben ser 1.

Ahora, ajustemos el modelo loglineal de independencia de dos vías al ejemplo de los esquiadores. Para ello, usaremos la función `glm()` convirtiendo nuestra tabla en un dataframe.

```
ski.data<-ski.data<-data.frame("Treatment"=c("Placebo","Placebo","VitaminC","VitaminC"),
                               "Cold"=c("Cold","NoCold","Cold","NoCold"),
                               "Freq"=c(31,109,17,122))
ski.data
```

```
##   Treatment   Cold Freq
## 1   Placebo   Cold   31
## 2   Placebo NoCold  109
## 3  VitaminC   Cold   17
## 4  VitaminC NoCold  122
```

Al aplicar `glm()` se requiere especificar `ski.data$Freq` como la variable que contiene los conteos de las celdas. En este modelo tenemos dos efectos principales sin el término de interacción. Es decir, `ski.data$Treatment+ski.data$Cold`. Finalmente, debemos especificar la distribución, en el caso de los modelos loglineales se debe definir `family=poisson()`.

```
ski.ind<-glm(ski.data$Freq~ski.data$Treatment+ski.data$Cold, family=poisson())

#### to view the model and the relevant statistics

ski.ind
```

```
##
## Call:  glm(formula = ski.data$Freq ~ ski.data$Treatment + ski.data$Cold,
##     family = poisson())
##
## Coefficients:
##                (Intercept)  ski.data$TreatmentVitaminC
##                   3.181632                   -0.007168
##        ski.data$ColdNoCold
##                   1.571217
##
## Degrees of Freedom: 3 Total (i.e. Null);  1 Residual
## Null Deviance:       135.5
## Residual Deviance: 4.872     AIC: 34
```

Otra forma de ver el modelo:

```
summary(ski.ind)
```

```
##
## Call:
## glm(formula = ski.data$Freq ~ ski.data$Treatment + ski.data$Cold,
##     family = poisson())
##
## Deviance Residuals:
##       1        2        3        4
##  1.3484  -0.6487  -1.4918   0.6382
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 3.181632   0.156179  20.372   <2e-16 ***
## ski.data$TreatmentVitaminC -0.007168   0.119738  -0.060    0.952
## ski.data$ColdNoCold         1.571217   0.158626   9.905   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 135.4675  on 3  degrees of freedom
## Residual deviance:   4.8717  on 1  degrees of freedom
## AIC: 34.004
##
## Number of Fisher Scoring iterations: 4
```

1.4 Modelo Saturado para tablas de dos vías

En este caso, también ajustaremos el término *interacción*, es decir, la asociación entre A y B.

1.4.1 Estructura del modelo

$$\log(\mu_{ij}) = \lambda + \lambda_i^A + \lambda_j^B + \lambda_{ij}^{AB}$$

Aquí, λ_{ij}^{AB} representa la interacción/asociación entre dos variables.

Ahora, añadimos un término de interacción al modelo `ski.data$Treatment*ski.data$Cold`:

```r
ski.sat<-glm(ski.data$Freq~ski.data$Treatment*ski.data$Cold, family=poisson())
ski.sat
```

```
##
## Call:  glm(formula = ski.data$Freq ~ ski.data$Treatment * ski.data$Cold,
##     family = poisson())
##
## Coefficients:
##                                       (Intercept)
##                                            3.4340
##                         ski.data$TreatmentVitaminC
##                                           -0.6008
##                               ski.data$ColdNoCold
##                                            1.2574
## ski.data$TreatmentVitaminC:ski.data$ColdNoCold
##                                            0.7134
##
## Degrees of Freedom: 3 Total (i.e. Null);  0 Residual
## Null Deviance:       135.5
## Residual Deviance: -5.773e-15    AIC: 31.13
```

```r
anova(ski.sat)
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: ski.data$Freq
##
## Terms added sequentially (first to last)
##
##
##                                    Df Deviance Resid. Df Resid. Dev
## NULL                                                    3    135.468
## ski.data$Treatment                  1    0.004         2    135.464
## ski.data$Cold                       1  130.592         1      4.872
## ski.data$Treatment:ski.data$Cold    1    4.872         0      0.000
```

1.5 Modelos Loglineales para tres variables

Para tablas de tres vías, existen múltiples modelos que se pueden probar y, luego, ajustar. Los modelos loglineales son:

Tipo	Modelo Loglineal	Notación
I. mutua	$\log(\mu_{ijk}) = \lambda + \lambda_i^A + \lambda_j^B + \lambda_k^C$	(A, B, C)
I. conjunta	$\log(\mu_{ijk}) = \lambda + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB}$	(AB, C)
I. condicional	$\log(\mu_{ijk}) = \lambda + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB} + \lambda_{jk}^{BC}$	(AB, BC)
Asociación Homogénea	$\log(\mu_{ijk}) = \lambda + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB} + \lambda_{jk}^{BC} + \lambda_{ik}^{AC}$	(AB, BC, AC)
Saturado	$\log(\mu_{ijk}) = \lambda + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB} + \lambda_{jk}^{BC} + \lambda_{ik}^{AC} + \lambda_{ijk}^{ABC}$	(ABC)

1.5.1 Modelo Loglineal Saturado

Este modelo es el modelo por omisión que sirve para pruebas de bondad de ajuste de los otros modelos. Recordemos que el modelo saturado tiene el máximo número de parámetros y ajustar un modelo saturado es lo mismo que estimar parámetros de máxima verosimilitud de distribuciones apropiadas para cada celda de la tabla de contingencia.

1.5.2 Estructura del modelo

$$\log(\mu_{ijk}) = \lambda + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB} + \lambda_{ik}^{AC} + \lambda_{jk}^{BC} + \lambda_{ijk}^{ABC}$$

Las restricciones implican

$$\sum_i \lambda_i^A = \sum_j \lambda_j^B = \sum_i \sum_j \lambda_{ij}^{AB} = \sum_i \sum_j \sum_k \lambda_{ijk}^{ABC} = 0$$

El modelo saturado tiene un ajuste perfecto, $G^2 = 0$, $g.l. = 0$ con $g.l.$ = número de celdas - el número de parámetros únicos en el modelo. El modelo saturado es el modelo más complejo posible. La selección del modelo es relevante al comparar con modelos más simples.

1.5.3 Ejemplo. Consumo de alcohol, cigarro y marihuana

La siguiente tabla muestra los resultados de un encuesta donde se le preguntó a estudiantes del último año de una preparatoria en Ohio si alguna vez habían consumido alcohol, cigarro o marihuana.

Alcohol	Cigarro	Marihuana (Si)	Marihuana (No)
Si	Si	911	538
	No	44	456
No	Si	3	43
	No	2	279

```
freq<-c(911,3,44,2,538,43,456,279)
nombres<-list(A=c("Si","No"),C=c("Si","No"),M=c("Si","No"))
drogas.tab<-array(freq,c(2,2,2),dimnames = nombres)
drogas.tab
```

```
## , , M = Si
##
##     C
## A     Si No
##   Si 911 44
##   No   3  2
##
## , , M = No
##
##     C
## A     Si  No
##   Si 538 456
```

```
##   No  43 279
```

En este ejemplo, usaremos el paquete `MASS` y la función `step` para encontrar el modelo más parsimonioso, es decir, aquel que ajuste los datos lo mejor posible con el menor número de parámetros.

```
library("MASS")
sat<-loglm(~ A*C*M, data=drogas.tab)
stp<-step(sat, direction = "backward",test="Chisq")
```

```
## Start:  AIC=16
## ~A * C * M
##
##         Df    AIC     LRT Pr(>Chi)
## - A:C:M  1 14.374 0.37399   0.5408
## <none>     16.000
##
## Step:  AIC=14.37
## ~A + C + M + A:C + A:M + C:M
##
##        Df    AIC    LRT  Pr(>Chi)
## <none>     14.37
## - A:M   1 104.02  91.64 < 2.2e-16 ***
## - A:C   1 199.75 187.38 < 2.2e-16 ***
## - C:M   1 509.37 497.00 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
stp
```

```
## Call:
## loglm(formula = ~A + C + M + A:C + A:M + C:M, data = drogas.tab,
##     evaluate = FALSE)
##
## Statistics:
##                       X^2 df  P(> X^2)
## Likelihood Ratio 0.3739859  1 0.5408396
## Pearson          0.4010998  1 0.5265218
```

Enseguida, podemos ver el resumen del procedimiento

```
summary(stp)
```

```
## Formula:
## ~A + C + M + A:C + A:M + C:M
## attr(,"variables")
## list(A, C, M)
## attr(,"factors")
##   A C M A:C A:M C:M
## A 1 0 0   1   1   0
## C 0 1 0   1   0   1
## M 0 0 1   0   1   1
```

```
## attr(,"term.labels")
## [1] "A"   "C"   "M"   "A:C" "A:M" "C:M"
## attr(,"order")
## [1] 1 1 1 2 2 2
## attr(,"intercept")
## [1] 1
## attr(,"response")
## [1] 0
## attr(,".Environment")
## <environment: R_GlobalEnv>
##
## Statistics:
##                         X^2 df  P(> X^2)
## Likelihood Ratio 0.3739859  1 0.5408396
## Pearson          0.4010998  1 0.5265218
```

2 Estimación por Sistemas Múltiples

En esta sección se ilustran los pasos para realizar una estimación del tamaño de una población a partir de listados de víctimas incompletos. El conjunto de datos utilizado representa una población de 10000 individuos. Estas notas han sido originalmente elaboradas por Maarten Cruyff, consultor de UNODC.

2.1 Planteamiento

Comenzamos cargando los datos del archivo `data.csv`.

```
d<-read.csv("data.csv")
head(d)
```

```
##   A B C X1 X2 Freq
## 1 0 0 0  1  1   NA
## 2 1 0 0  1  1   70
## 3 0 1 0  1  1   62
## 4 1 1 0  1  1   20
## 5 0 0 1  1  1  244
## 6 1 0 1  1  1    6
```

Supóngase que se ha hecho la identificación de coincidencias de tres listas incompletas de víctimas de trata A, B, C donde `0` significa que no fue identificada o registrada y `1` que sí se identificó. Las variables `X1,X2` contienen características de la población, *covariados*, por ejemplo, `X1` es el sexo donde `1` es hombre y `2` es mujer y `X2` es la edad donde `1` es menor de edad y `2` es adulto. Las frecuencias en los renglones con ceros para A, B y C tienen `NA` porque son las celdas que corresponden a la víctimas que no se registraron en alguna de las tres listas. Estos son los valores que se desean estimar.

2.2 Estimación con dos sistemas

Primero, consideraremos el caso en el que se tienen dos listas A, B sin covariados.

```
AB <- subset(d, select = c(A, B, Freq), subset = A==1 | B==1)
d_AB <- as.data.frame(xtabs(Freq ~ A + B, data = AB))
d_AB
```

```
##   A B Freq
## 1 0 0    0
## 2 1 0  321
## 3 0 1 4585
## 4 1 1  381
```

La frecuencia de la celda n_{00} es cero pues se trata de las víctimas que no fueron registradas en alguna de las dos listas. A las celdas con esta característica las llamaremos *ceros estructurales*. Ahora, formulamos el modelo para estimar su valor.

El modelo loglineal saturado (AB) para dos listas A y B es

$$\log m_{ij} = \lambda_0 + \lambda_i^A + \lambda_j^B + \lambda_{ij}^{AB},$$

para $i, j \in \{0, 1\}$, donde m_{ij} es la frecuencia ajustada de la celda ij. Recordemos que la notación (AB) representa el modelo jerárquico que contiene el efecto de interacción, así como los efectos principales de A y B.

Desafortunadamente, este modelo está sobreparametrizado porque tiene 4 parámetros y sólo 3 frecuencias observadas. Para resolver esto, hacemos que el parámetro de interacción sea cero. Es decir, usaremos el modelo (A, B):

$$\log m_{ij} = \lambda_0 + \lambda_i^A + \lambda_j^B,$$

el cual supone que la probabilidad de inclusión en una lista es independiente de la probabilidad de inclusión de otra lista. Ajustamos el modelo con la función `glm()` al conjunto de datos `d_AB` sin el cero estructural:

```
d_AB.glm <- glm(Freq ~ A + B, family = poisson, data = d_AB,
                subset = !(A == 0 & B == 0))
```

Veamos el resumen de la estimación:

```
summary(d_AB.glm)
```

```
##
## Call:
## glm(formula = Freq ~ A + B, family = poisson, data = d_AB, subset = !(A ==
##     0 & B == 0))
##
## Deviance Residuals:
## [1]  0  0  0
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  8.25919    0.07719 107.000   <2e-16 ***
## A1          -2.48775    0.05332 -46.659   <2e-16 ***
## B1           0.17136    0.07576   2.262   0.0237 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 6.5075e+03  on 2  degrees of freedom
## Residual deviance: 1.2657e-13  on 0  degrees of freedom
## AIC: 31.659
##
## Number of Fisher Scoring iterations: 2
```

Para estimar la frecuencia de la celda m_{00}, utilizamos la siguiente instrucción:

```
d_AB.pred <- predict(d_AB.glm, newdata = d_AB, type = "response")
```

La suma de estas frecuencias nos da el tamaño de la población estimada

```
Nhat_AB <- sum(d_AB.pred)
```

2.2.1 Ejercicio. Ejecute el mismo análisis para las parejas A, C y B, C.

2.3 Estimación con dos sistemas y covariados

La inclusión de los covariados en el modelo permitirá que la inclusión de probabilidades varíe sobre los niveles de estos. Nuevamente, sólo consideraremos las listas A y B añadiendo la información de X_1 y X_2:

```
ABX   <- subset(d, select = c(A, B, X1, X2, Freq), subset = !(A == 0 & B == 0))
d_ABX <- as.data.frame(xtabs(Freq ~ A + B + X1 + X2, data = ABX))
d_ABX
```

```
##    A B X1 X2 Freq
## 1  0 0  1  1    0
## 2  1 0  1  1   76
## 3  0 1  1  1  203
## 4  1 1  1  1   30
## 5  0 0  2  1    0
## 6  1 0  2  1  160
## 7  0 1  2  1 3226
## 8  1 1  2  1  236
## 9  0 0  1  2    0
## 10 1 0  1  2   33
## 11 0 1  1  2   89
## 12 1 1  1  2   18
## 13 0 0  2  2    0
## 14 1 0  2  2   52
## 15 0 1  2  2 1067
## 16 1 1  2  2   97
```

En este conjunto de datos hay 4 ceros estructurales, uno para cada combinación de los covariados. Entonces el modelo ahora es (AX_1X_2, BX_1X_2), o bien:

$$\log m_{ijkl} = \lambda_0 + \lambda_i^A + \lambda_j^B + \lambda_k^{X_1} + \lambda_l^{X_2} + \lambda_{ik}^{AX_1} + \lambda_{il}^{AX_2} + \lambda_{ik}^{BX_1} + \lambda_{il}^{BX_2} + \lambda_{kl}^{X_1X_2} + \lambda_{ikl}^{AX_1X_2} + \lambda_{jkl}^{BX_1X_2}.$$

Para encontrar el modelo más parsimonioso, usaremos la función `step()`. Esta función requiere que se ajuste el modelo de los efectos principales:

```
d_ABX.main <- glm(Freq ~ ., family=poisson, data = d_ABX, subset = !(A==0 & B==0))
```

Enseguida, aplicamos el procedimiento *stepwise* al modelo anterior. Para ello, necesitamos especificar el alcance de los modelos en los cuales buscar el más parsimonioso. Esto es, (AX_1X_2, BX_1X_2)

```
d_ABX.step <- step(d_ABX.main, scope = ~ (A + B)*X1*X2)
```

```
## Start:  AIC=301.7
## Freq ~ A + B + X1 + X2
##
##         Df Deviance    AIC
## + B:X1   1     29.3  121.4
## + A:X1   1     57.8  149.9
## + X1:X2  1    204.0  296.1
## + A:X2   1    208.2  300.3
## <none>        211.6  301.7
## + B:X2   1    211.5  303.6
## - B      1    216.7  304.8
## - X2     1   1520.7 1608.8
## - A      1   4407.4 4495.6
## - X1     1   4467.7 4555.8
##
```

```
## Step:  AIC=121.4
## Freq ~ A + B + X1 + X2 + B:X1
##
##         Df Deviance    AIC
## + A:X1   1     11.5  105.6
## + X1:X2  1     21.7  115.8
## + A:X2   1     25.9  120.0
## <none>          29.3  121.4
## + B:X2   1     29.2  123.3
## - B:X1   1    211.6  301.7
## - X2     1   1338.4 1428.5
## - A      1   4225.1 4315.3
##
## Step:  AIC=105.63
## Freq ~ A + B + X1 + X2 + B:X1 + A:X1
##
##         Df Deviance     AIC
## + X1:X2  1     3.93  100.05
## + A:X2   1     8.14  104.27
## <none>        11.51  105.63
## + B:X2   1    11.38  107.51
## - A:X1   1    29.28  121.40
## - B:X1   1    57.80  149.92
## - X2     1  1320.63 1412.76
##
## Step:  AIC=100.05
## Freq ~ A + B + X1 + X2 + B:X1 + A:X1 + X1:X2
##
##         Df Deviance    AIC
## <none>         3.928 100.05
## + A:X2   1    2.199 100.32
## + B:X2   1    3.835 101.96
## - X1:X2  1   11.506 105.63
## - A:X1   1   21.702 115.83
## - B:X1   1   50.219 144.34
```

```
d_ABX.step
```

```
##
## Call:  glm(formula = Freq ~ A + B + X1 + X2 + B:X1 + A:X1 + X1:X2, family = poisson,
##     data = d_ABX, subset = !(A == 0 & B == 0))
##
## Coefficients:
## (Intercept)           A1           B1          X12          X22       B1:X12
##      6.1232      -1.8056      -0.8201       1.5005      -0.7917       1.2717
##      A1:X12      X12:X22
##     -0.7510      -0.2998
##
## Degrees of Freedom: 11 Total (i.e. Null);  4 Residual
## Null Deviance:       12280
## Residual Deviance: 3.928     AIC: 100.1
```

El tamaño de la población estimada es:

```r
d_ABX.pred <- predict(d_ABX.step, newdata = d_ABX, type = "response")
Nhat_ABX <- sum(d_ABX.pred)
```

También, podemos obtener los valores estimados para los niveles de los covariados:

```r
xtabs(d_ABX.pred ~ X1, data=d_ABX)
```

```
## X1
##        1        2
## 1112.083 7571.081
```

2.4 Estimación con tres sistemas y covariados

A continuación, añadiremos una tercera lista al modelo. Recordemos que la principal ventaja es que esto permitirá interacciones por parejas de listas. Por lo tanto, utilizaremos toda la información del conjunto `d` replicando los mismos pasos del ejercicio anterior.

```r
xtabs(Freq ~ A + B + C, data=d)
```

```
## , , C = 0
##
##    B
## A      0    1
##   0    0  491
##   1  301  182
##
## , , C = 1
##
##    B
## A      0    1
##   0 1430 4094
##   1   20  199
```

Ajustemos el modelo de efectos principales `Freq~A+B+C+X1+X2` al conjunto `d` excluyendo los ceros estructurales con el comando `subset` y guardamos el objeto como `d.main`.

Ahora, estimemos el modelo más parsimonioso para `d` usando la función `step()` y estableciendo como alcance el modelo de asociaciones homégeneas en combinación con los covariados. Guarde el modelo como `d.step`.

```
## Start:  AIC=2196.08
## Freq ~ A + B + C + X1 + X2
##
##         Df Deviance    AIC
## + A:C    1    981.5 1160.7
## + B:C    1   1067.6 1246.8
## + B:X1   1   1517.6 1696.9
## + C:X1   1   1927.6 2106.8
## + A:B    1   1928.5 2107.7
## + A:X1   1   1960.5 2139.7
## + B:X2   1   1990.5 2169.7
## + X1:X2  1   1993.9 2173.2
```

```
## <none>       2018.9 2196.1
## + A:X2   1   2018.3 2197.6
## + C:X2   1   2018.5 2197.8
## - B      1   3016.7 3191.9
## - X2     1   3450.9 3626.1
## - C      1   4219.6 4394.9
## - X1     1   6244.3 6419.5
## - A      1   7111.3 7286.5
##
## Step:  AIC=1160.69
## Freq ~ A + B + C + X1 + X2 + A:C
##
##         Df Deviance    AIC
## + B:X1   1    480.2  661.5
## + B:C    1    713.5  894.8
## + A:B    1    869.6 1050.8
## + C:X1   1    890.2 1071.4
## + A:X1   1    923.1 1104.3
## + B:X2   1    953.1 1134.3
## + X1:X2  1    956.6 1137.8
## <none>       981.5 1160.7
## + A:X2   1    980.9 1162.2
## + C:X2   1    981.2 1162.4
## - A:C    1   2018.9 2196.1
## - B      1   2214.5 2391.8
## - X2     1   2413.5 2590.7
## - X1     1   5206.9 5384.1
##
## Step:  AIC=661.46
## Freq ~ A + B + C + X1 + X2 + A:C + B:X1
##
##         Df Deviance     AIC
## + B:C    1   212.31  395.54
## + C:X1   1   313.22  496.44
## + A:B    1   368.34  551.56
## + A:X1   1   427.89  611.12
## + B:X2   1   451.84  635.06
## + X1:X2  1   455.33  638.55
## <none>       480.24  661.46
## + A:X2   1   479.72  662.94
## + C:X2   1   479.93  663.15
## - B:X1   1   981.47 1160.69
## - A:C    1  1517.64 1696.86
## - X2     1  1912.28 2091.50
##
## Step:  AIC=395.54
## Freq ~ A + B + C + X1 + X2 + A:C + B:X1 + B:C
##
##         Df Deviance     AIC
## + C:X1   1   133.89  319.11
## + A:B    1   174.10  359.32
## + B:X2   1   183.91  369.13
## + X1:X2  1   187.40  372.62
## + A:X1   1   201.39  386.62
```

```
## <none>       212.31  395.54
## + A:X2   1   211.79  397.01
## + C:X2   1   212.00  397.22
## - B:C    1   480.24  661.46
## - A:C    1   566.35  747.57
## - B:X1   1   713.54  894.76
## - X2     1  1644.35 1825.57
##
## Step:  AIC=319.11
## Freq ~ A + B + C + X1 + X2 + A:C + B:X1 + B:C + C:X1
##
##          Df Deviance     AIC
## + A:B     1    95.68  282.90
## + B:C:X1  1    95.87  283.09
## + B:X2    1   105.49  292.71
## + X1:X2   1   108.98  296.20
## + A:X1    1   121.40  308.62
## <none>        133.89  319.11
## + A:X2    1   133.37  320.59
## + C:X2    1   133.57  320.80
## - C:X1    1   212.31  395.54
## - B:C     1   313.22  496.44
## - A:C     1   487.92  671.14
## - B:X1    1   622.27  805.49
## - X2      1  1565.93 1749.15
##
## Step:  AIC=282.9
## Freq ~ A + B + C + X1 + X2 + A:C + B:X1 + B:C + C:X1 + A:B
##
##          Df Deviance     AIC
## + B:C:X1  1    57.66  246.88
## + B:X2    1    67.28  256.50
## + X1:X2   1    70.76  259.99
## + A:X1    1    88.65  277.87
## <none>         95.68  282.90
## + A:X2    1    95.16  284.38
## + C:X2    1    95.36  284.58
## - A:B     1   133.89  319.11
## - C:X1    1   174.10  359.32
## - B:C     1   251.63  436.85
## - A:C     1   387.46  572.68
## - B:X1    1   584.05  769.27
## - X2      1  1527.72 1712.94
##
## Step:  AIC=246.88
## Freq ~ A + B + C + X1 + X2 + A:C + B:X1 + B:C + C:X1 + A:B +
##     B:C:X1
##
##          Df Deviance     AIC
## + B:X2    1    29.25  220.47
## + X1:X2   1    32.74  223.96
## <none>         57.66  246.88
## + A:X1    1    55.87  247.09
## + A:X2    1    57.13  248.35
```

```
## + C:X2    1    57.34  248.56
## - B:C:X1  1    95.68  282.90
## - A:B     1    95.87  283.09
## - A:C     1   349.43  536.65
## - X2      1  1489.70 1676.92
##
## Step:  AIC=220.47
## Freq ~ A + B + C + X1 + X2 + A:C + B:X1 + B:C + C:X1 + A:B +
##     B:X2 + B:C:X1
##
##          Df Deviance    AIC
## + X1:X2   1    16.45 209.67
## <none>         29.25 220.47
## + A:X1    1    27.47 220.69
## + C:X2    1    28.60 221.82
## + A:X2    1    29.24 222.46
## - B:X2    1    57.66 246.88
## - B:C:X1  1    67.28 256.50
## - A:B     1    67.46 256.69
## - A:C     1   321.03 510.25
##
## Step:  AIC=209.67
## Freq ~ A + B + C + X1 + X2 + A:C + B:X1 + B:C + C:X1 + A:B +
##     B:X2 + X1:X2 + B:C:X1
##
##           Df Deviance    AIC
## <none>         16.447 209.67
## + A:X1     1   14.663 209.88
## + C:X2     1   14.926 210.15
## + A:X2     1   16.374 211.60
## + B:X1:X2  1   16.418 211.64
## - X1:X2    1   29.252 220.47
## - B:X2     1   32.741 223.96
## - B:C:X1   1   54.470 245.69
## - A:B      1   54.659 245.88
## - A:C      1  308.223 499.44
```

Enseguida, guardamos los valores de la predicción como `d.pred` y el tamaño de la población como `Nhat`.

```
## [1] 9539.201
```

2.5 Intervalo de confianza

Por último, calcularemos un intervalo de confianza alrededor de $\hat{N}$ con el método bootstrap el cual simulará una muestra aleatoria de una distribución multinomial. Luego, se filtrarán los ceros estructurales de los datos y se ajustará el modelo para obtener las predicciones del conjunto completo, así como el tamaño de la población. Repetiremos este proceso 1000 veces y se calculará los percentiles 2.5 y 97.5.

```r
set.seed(10)
boot.nhats <- NULL
for(i in 1:1000){
  d.sample   <- sample(1:32, size=Nhat, replace=TRUE, prob=d.pred/Nhat)
  d.boot     <- as.data.frame(table(d[d.sample, -6]))
```

```
  boot.glm   <- glm(d.step$formula, family=poisson, data=d.boot,
                    subset=!(A==0&B==0&C==0))
  boot.pred  <- predict(boot.glm, newdata=d.boot, type="response")
  boot.nhats <- c(boot.nhats, sum(boot.pred))
}
quantile(boot.nhats, c(.025, .975))
```

```
##      2.5%     97.5%
##  8519.113 11723.462
```