Entrega 3 - Inferencia bayesiana - Bioestadística

Francisco Javier López Carbonell

19/12/2023

ÍNDICE DE CONTENIDOS

Análisis bayesiano de datos experimentales

Se realizó un estudio en el Baystate Medical Center, Springfield Massachusetts, con el objetivo de identificar factores asociados con el riesgo de tener un bebé con bajo peso al nacer (menos de 2500 gr). Se tomaron datos de 189 mujeres embarazadas, 59 de las cuales tuvieron un bebé con bajo peso al nacer. En la base de datos Lowbirthweight tenemos las siguientes variables:

-LOW: indicador de bajo peso al nacer. Valores 0 (no), 1(sí).

-age5c: edad de la madre en grupos de edad. Valores 1 (≤ 18), 2 ((18, 20]), 3 ((20, 25]), 4 ((25,30]) y 5 (> 30).

-smoke: fumar durante el embarazo. Valores 1 (no), 2 (sí).

-ptl: indicador de partos previos. Valores 1 (no), 2 (sí).

-race: raza de la madre. Valores 1(blanca), 2(negra), 3(otra).

-ui: irritabilidad del útero. Valores 1 (no), 2 (sí).

Usaremos como distribución a priori una uniforme. El modelo está especificado en el fichero modelo.txt, donde podrás observar que algunos parámetros se han especificado a valor 0. Esto se hace para que el modelo sea identificable.

Para los siguientes apartados: utiliza 50000 simulaciones, tres cadenas, un periodo de quemado de 5000 simulaciones y un thin de 10.

Primero se cargan en R los datos del modelo, usando para ello la libreria readxl.

```
library(readxl)

## Warning: package 'readxl' was built under R version 4.3.2

Lowbirthweight <-
read_excel("C:/Users/Javi2/OneDrive/Escritorio/Entrega3-
Bioestadística/Lowbirthweight.xlsx")
attach(Lowbirthweight)
```

Cargamos la librería que conecta R con Winbugs.

Definimos el conjunto de datos que se le pasará a WinBugs para hacer el analisis bayesiano, deben ser tipo lista las variables y debe estar definido n en el conjunto.

```
datos
=list(LOW=Lowbirthweight$LOW,smoke=Lowbirthweight$smoke,age5c=Lowbirthwei
ght$age5c,ptl=Lowbirthweight$ptl,race=Lowbirthweight$race,ui=Lowbirthweig
ht$ui,n=length(LOW))
```

Con la librería R2WinBUGS de R se puede realizar MCMC con Winbugs desde R. Una vez obtenidos los datos, los datos iniciales y el modelo, se utiliza la función bugs de la librería para realizar la simulación. El resultado se guarda en la variable resultado.

```
library(R2WinBUGS)

## Warning: package 'R2WinBUGS' was built under R version 4.3.2

## Loading required package: coda

## Warning: package 'coda' was built under R version 4.3.2

## Loading required package: boot

#Inicializamos los parámetros con la siguiente función proporcionada.

iniciales=function(){
  list(alpha=rnorm(1,0,1),beta=c(NA,rnorm(4,0,1)),
       gamma=c(NA,rnorm(1,0,1)),delta=c(NA,rnorm(1,0,1)),
       phi=c(NA,rnorm(2,0,1)),omega=c(NA,rnorm(1,0,1)))
}
```

```r
#Definimos el modelo que va a usar Winbugs y el directorio de winbugs
para que R pueda llamarlo y ejecutarlo.

model="modelo.txt"
directorio.winbugs="C:/Users/Javi2/OneDrive/Escritorio/MASTER
BIOINFORMÁTICA UMU 2023-2024/Bioestadística/Temario/Inferencia
bayesiana/winbugs143_unrestricted/winbugs14_full_patched/WinBUGS14"

resultado=bugs(data=datos,inits=iniciales,model.file=model,
parameters.to.save=c("alpha","beta", "gamma", "delta",
"phi","omega"),n.iter=50000,n.burnin=5000,n.thin=10,n.chain=3,
bugs.directory=directorio.winbugs,DIC=F, debug = T)

attach_res <- attach.bugs(resultado)
print(resultado,digit=3)

## Inference for Bugs model at "modelo.txt", fit using WinBUGS,
##  3 chains, each with 50000 iterations (first 5000 discarded), n.thin =
10
##  n.sims = 13500 iterations saved
##            mean    sd   2.5%    25%    50%    75%  97.5%  Rhat n.eff
## alpha    -2.024 0.540 -3.098 -2.379 -2.016 -1.654 -0.999 1.001  7200
## beta[2]  -0.200 0.572 -1.324 -0.579 -0.195  0.184  0.919 1.001 14000
## beta[3]   0.087 0.500 -0.864 -0.251  0.080  0.426  1.081 1.001  9500
## beta[4]  -0.237 0.590 -1.414 -0.633 -0.237  0.164  0.911 1.001 14000
## beta[5]  -1.205 0.849 -2.983 -1.748 -1.177 -0.611  0.354 1.001  4500
## gamma[2]  1.019 0.409  0.232  0.743  1.014  1.294  1.829 1.001 14000
## delta[2]  1.311 0.475  0.390  0.990  1.307  1.632  2.243 1.001 14000
## phi[2]    1.108 0.522  0.084  0.756  1.108  1.460  2.126 1.001 14000
## phi[3]    1.002 0.440  0.163  0.702  0.998  1.294  1.875 1.001 14000
## omega[2]  0.720 0.486 -0.237  0.393  0.718  1.046  1.667 1.001 14000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence,
Rhat=1).
```

APARTADOS

1. Calcular la media a posteriori y un IC95% para los par´ametros α, β2, β3, β4, β5, γ2, δ2, ϕ2, ϕ3, y ω2.

Media a posteriori de los parámetros.

```r
media <- resultado$mean
media

## $alpha
## [1] -2.024431
##
```

```
## $beta
## [1] -0.19974898  0.08722023 -0.23676402 -1.20530191
##
## $gamma
## [1] 1.019484
##
## $delta
## [1] 1.311458
##
## $phi
## [1] 1.108157 1.001655
##
## $omega
## [1] 0.7196371
```

Intervalo de credibilidad al 95% en la estimación de los parámetros y gráfico de densidad.

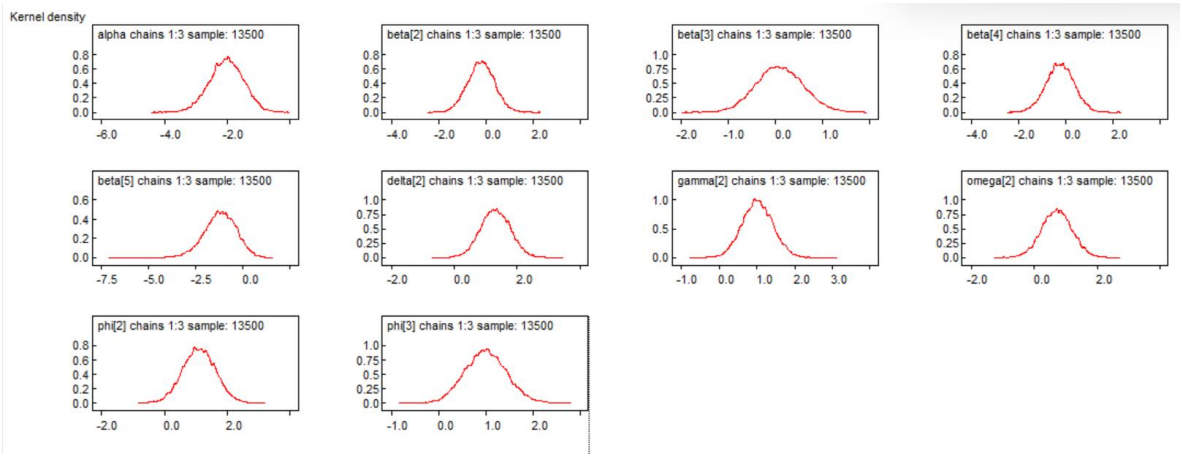
```
# Crear un dataframe para los parámetros beta y su IC95%
beta_quantiles <- data.frame(matrix(ncol = 3, nrow = 4))
colnames(beta_quantiles) <- c("Parameter", "per2.5", "per97.5")

for (i in 1:4) {
  variable_name <- paste('beta', i + 1, sep=' ')
  qb <- round(quantile(attach_res$beta[, i], probs = c(0.025, 0.975)),4)
  beta_quantiles[i, ] <- c(variable_name, qb[1], qb[2])
}

# Crear un dataframe para resto de parametros e intervalos de
credibilidad
other_quantiles <- data.frame(
  Variable = c("alpha", "gamma[2]", "delta[2]", "phi[2]", "phi[3]",
"omega[2]"),
  per2.5 = round(c(quantile(attach_res$alpha, probs = c(0.025,
0.975))[1],
            quantile(attach_res$gamma[2], probs = c(0.025, 0.975))[1],
            quantile(attach_res$delta[2], probs = c(0.025, 0.975))[1],
            quantile(attach_res$phi[2], probs = c(0.025, 0.975))[1],
            quantile(attach_res$phi[3], probs = c(0.025, 0.975))[1],
            quantile(attach_res$omega[2], probs = c(0.025,
0.975))[1]),4),
  per97.5 = round(c(quantile(attach_res$alpha, probs = c(0.025,
0.975))[2],
            quantile(attach_res$gamma[2], probs = c(0.025, 0.975))[2],
            quantile(attach_res$delta[2], probs = c(0.025, 0.975))[2],
            quantile(attach_res$phi[2], probs = c(0.025, 0.975))[2],
            quantile(attach_res$phi[3], probs = c(0.025, 0.975))[2],
            quantile(attach_res$omega[2], probs = c(0.025, 0.975))[2]),4)
)
```

```
# Mostrar los resultados en dos dataframe
print(beta_quantiles)
```

```
##   Parameter  per2.5 per97.5
## 1    beta 2 -1.3235  0.9191
## 2    beta 3 -0.8635   1.081
## 3    beta 4 -1.4135  0.9112
## 4    beta 5  -2.983  0.3537
```

```
print(other_quantiles)
```

```
##   Variable  per2.5 per97.5
## 1    alpha -3.0980 -0.9986
## 2 gamma[2]  1.2170  1.2170
## 3 delta[2]  1.0930  1.0930
## 4   phi[2]  1.8670  1.8670
## 5   phi[3]  1.1650  1.1650
## 6 omega[2] -0.5267 -0.5267
```

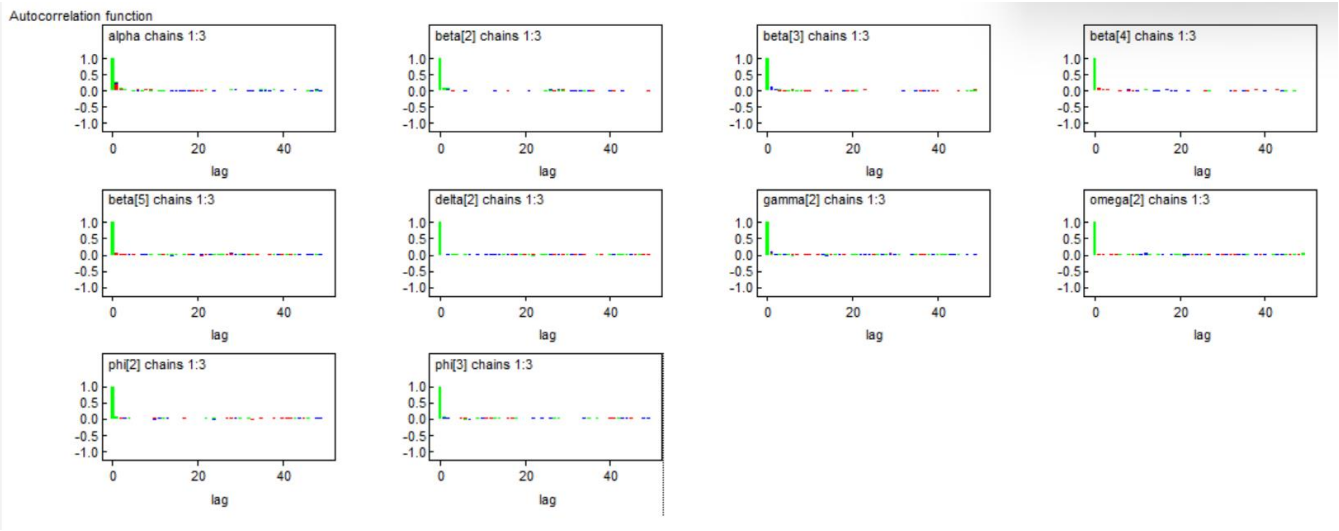


2. Valorar la convergencia obteniendo las autocorrelaciones para esos parámetros, así como el Rhat y el n.eff

Evaluación de convergencia

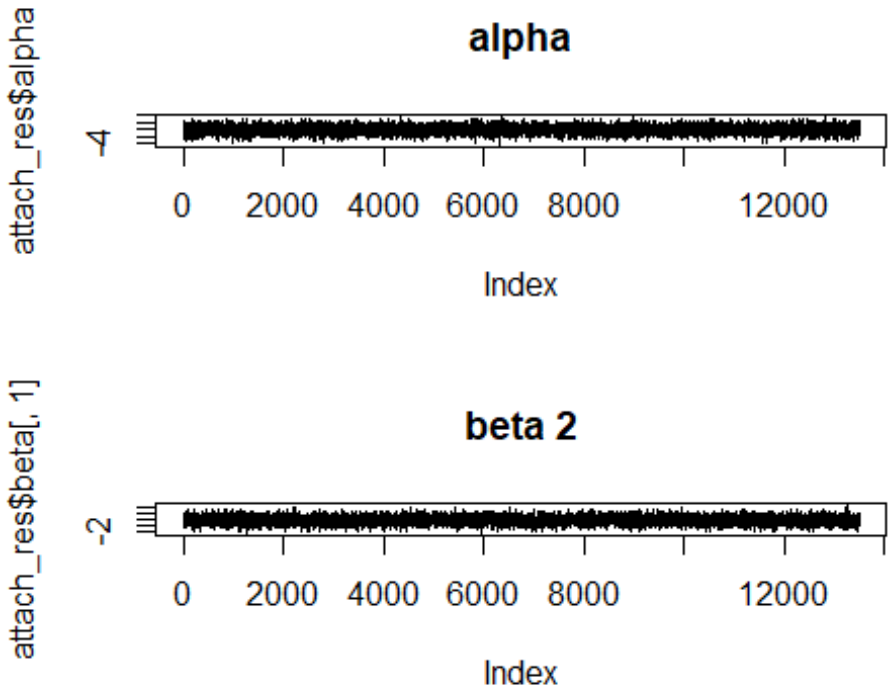
En primer lugar, aportamos las funciones de autocorrelación extraidas de Winbugs.Si la ACF baja rápidamente a cero, indica que la cadena de Markov ha convergido rápidamente a la distribución estacionaria. Esto es positivo y sugiere una buena convergencia. Una baja autocorrelación implica que las muestras generadas son más independientes entre sí. Cuanto menor sea la autocorrelación, más eficientes son las muestras para estimar parámetros.

En este caso, se puede observar que, para todos los parámetros bajan muy rápidamente por lo que el modelo converge rápidamente con estas especificaciones.

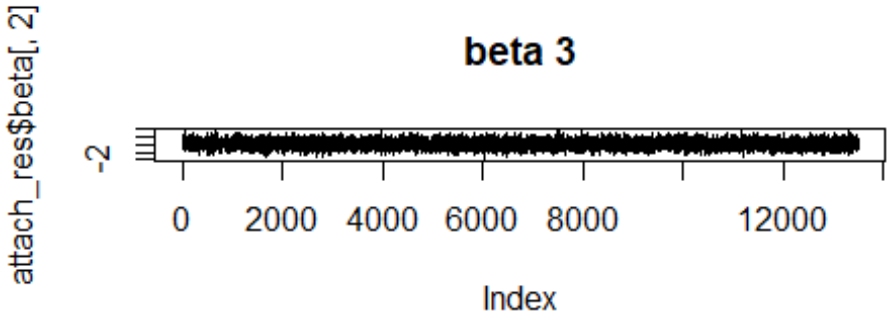


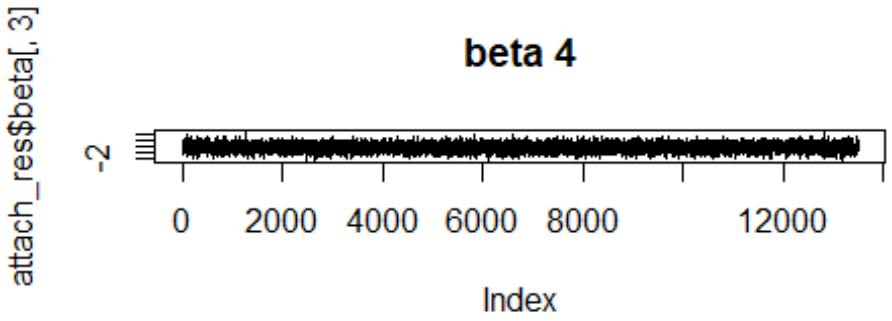
Gráficos de traza para cada parámetro

```
o <- par(mfrow=c(2,1))
plot(attach_res$alpha, type='l', main='alpha')
plot(attach_res$beta[,1], type='l', main='beta 2')
```

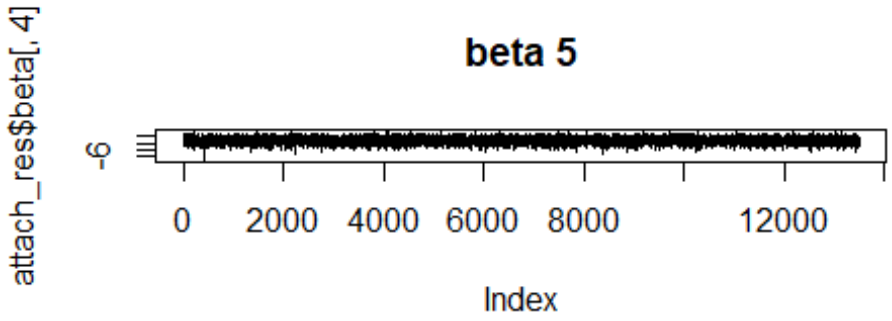


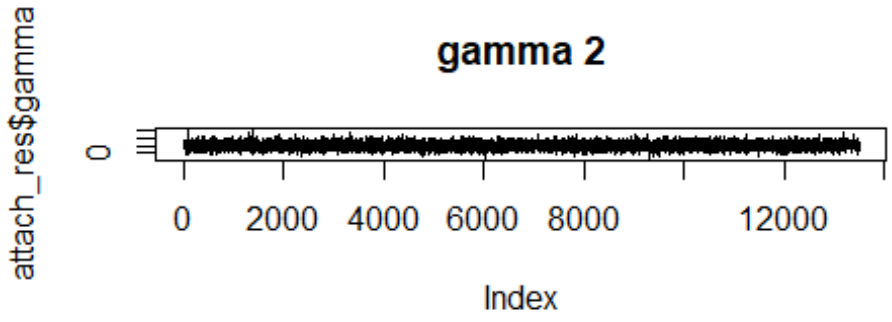

```
plot(attach_res$beta[,2], type='l', main='beta 3')
plot(attach_res$beta[,3], type='l', main='beta 4')
```



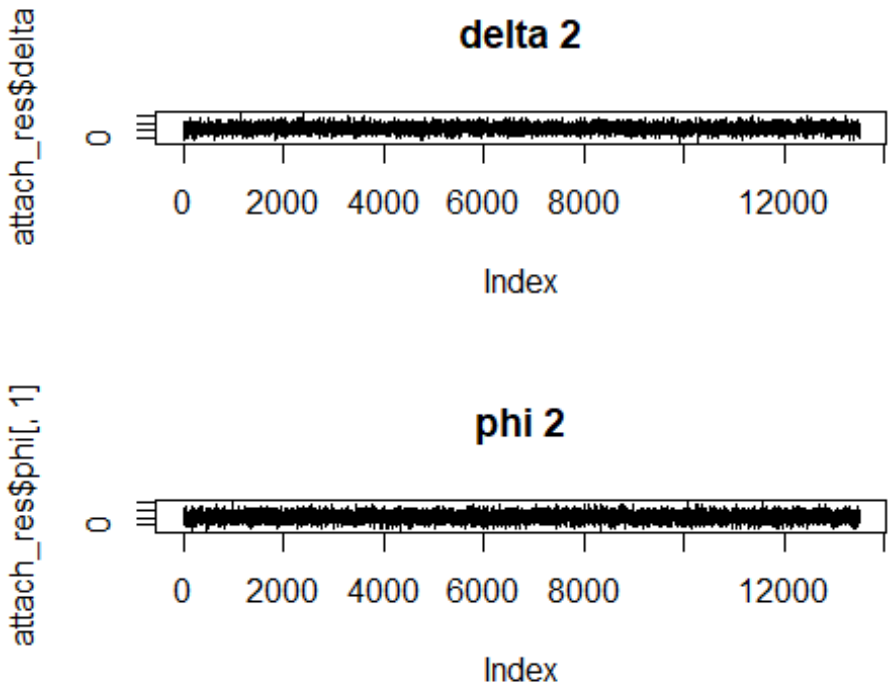



```r
plot(attach_res$beta[,4], type='l', main='beta 5')
plot(attach_res$gamma, type='l', main='gamma 2')
```

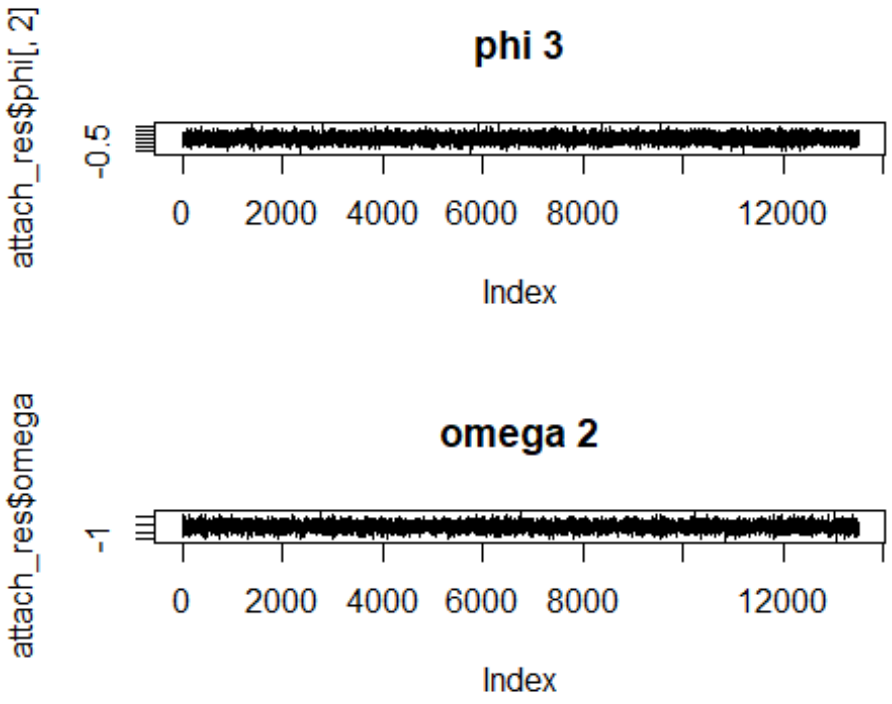




```r
plot(attach_res$delta, type='l', main='delta 2')
plot(attach_res$phi[,1], type='l', main='phi 2')
```




```r
plot(attach_res$phi[,2], type='l', main='phi 3')
plot(attach_res$omega, type='l', main='omega 2')
```



```r
par(o)
```

Además, atenderemos al valor de Rhat y n.eff, para validar la convergencia. En inferencia bayesiana, Rhat y n.eff son diagnósticos utilizados para evaluar la convergencia de las cadenas de Markov Monte Carlo (MCMC) en el contexto de muestreo de Markov Monte Carlo Hamiltoniano (HMC).

Si Rhat no es mas grande que 1.1, se considera que el metodo ha convergido para el parametro en cuestion. En este caso -> practicamente 1 (muy bueno) -> mientras no sobrepase el 1.1 se considera convergencia aceptable. n.eff (simulaciones efectivas) la mayoría de parámetros en torno a 14.000 simulaciones (simulaciones independientes) aunque algunos un poco más bajo.

Esto también se puede apreciar en el gráfico de la traza para cada parámetro. Nosotros hemos realizado 50.000 simulaciones independientes, que son bastantes y a partir de 100 se considera un valor de n.eff aceptable. Si n.eff no fuera mayor a 100, debemos hacer mas interacciones para que se aceptable.

3. Obtener la media a posteriori, el IC95% y la distribución a posteriori, para los odds ratio correspondientes a las categorías de la raza tomando como referencia la raza blanca: parámetros OR21 y OR31

```r
#Definir la categoría de referencia
ref_race <- 1  # Raza blanca

#Obtener las simulaciones de los parámetros de interés
dim(attach_res$phi)
```

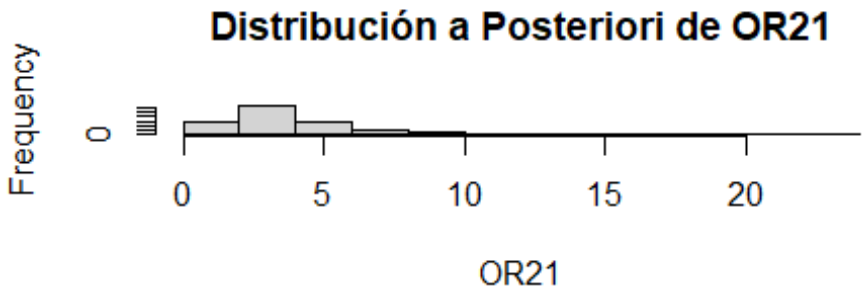
```
## [1] 13500     2
```

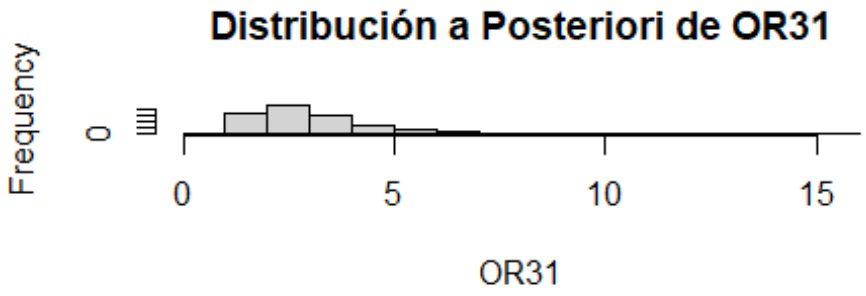
```r
OR21 <- exp(attach_res$phi[, 1])  # phi[2] es log(OR21)
OR31 <- exp(attach_res$phi[, 2])  # phi[3] es log(OR31)

# Calcular la media a posteriori de los dos parámetros
mean_OR21 <- mean(OR21)
mean_OR31 <- mean(OR31)

# Calcular intervalo de credibilidad del 95%
IC_OR21 <- quantile(OR21, c(0.025, 0.975))
IC_OR31 <- quantile(OR31, c(0.025, 0.975))

# Resumen de resultados
```

```r
summary_results <- data.frame(
  Parameter = c("OR21", "OR31"),
  Mean = c(mean_OR21, mean_OR31),
  IC_95_lower = c(IC_OR21[1], IC_OR31[1]),
  IC_95_upper = c(IC_OR21[2], IC_OR31[2])
)

print(summary_results)
```

```
##   Parameter     Mean IC_95_lower IC_95_upper
## 1      OR21 3.471097    1.087417    8.381275
## 2      OR31 3.002182    1.176537    6.517723
```

```r
# Histograma distribución a posteriori
par(mfrow = c(2, 1))
hist(OR21, main = "Distribución a Posteriori de OR21", xlab = "OR21")
hist(OR31, main = "Distribución a Posteriori de OR31", xlab = "OR31")
```





Un odds ratio mayor que 1, sugiere que la exposición está asociada con mayores probabilidades de ocurrencia del evento en comparación con el grupo de referencia. Cuanto mayor es el OR, mayor es la asociación.

El odds ratio promedio para la categoría de raza 2 (comparada con la raza blanca) es de 3.4. El intervalo de credibilidad del 95% indica que estamos razonablemente seguros de que el verdadero odds ratio está en el rango de 1 a 8. El odds ratio promedio para la categoría de raza 3 (comparada con la raza blanca) es de 3. El intervalo de credibilidad del

95% indica que estamos razonablemente seguros de que el verdadero odds ratio está en el rango de 1 a 6.5.

En ambos casos, un odds ratio mayor que 1 sugiere que hay una asociación positiva entre la categoría de raza específica y el evento de interés en comparación con la raza blanca.

La probabilidad de cuanto más probable se evalúa en el siguiente apartado, modificando el modelo y calculando las probabilidades de tener hijo de bajo de peso si todas fueran de raza blanca , todas fueran de raza negra o todas fueran de otra raza distinta a blanca o negra.

Concluimos por ahora que, habrá mayor probabilidad de concebir un niño con bajo si la madre es de raza negra u otra raza segun nuestros resultados.

4. Obtener la media a posteriori, el IC95% y la distribuciOn a posteriori, para los parámetros π1, π2, π3, π2/π1 y π3/π1.

```
library(R2WinBUGS)

#Inicializamos los parámetros con la siguiente función proporcionada.

iniciales=function(){
  list(alpha=rnorm(1,0,1),beta=c(NA,rnorm(4,0,1)),
       gamma=c(NA,rnorm(1,0,1)),delta=c(NA,rnorm(1,0,1)),
       phi=c(NA,rnorm(2,0,1)),omega=c(NA,rnorm(1,0,1)))
}

#Definimos el modelo que va a usar Winbugs y el directorio de winbugs
para que R pueda llamarlo y ejecutarlo.

model="modeloalt.txt"
directorio.winbugs="C:/Users/Javi2/OneDrive/Escritorio/MASTER
BIOINFORMÁTICA UMU 2023-2024/Bioestadística/Temario/Inferencia
bayesiana/winbugs143_unrestricted/winbugs14_full_patched/WinBUGS14"

resultado2=bugs(data=datos,inits=iniciales,model.file=model,
parameters.to.save=c("meanp1","meanp2", "meanp3", "meanp2p1",
"meanp3p1"),n.iter=50000,n.burnin=5000,n.thin=10,n.chain=3,
bugs.directory=directorio.winbugs,DIC=F)

attach_res2 <- attach.bugs(resultado2)
print(resultado2,digit=3)

## Inference for Bugs model at "modeloalt.txt", fit using WinBUGS,
##  3 chains, each with 50000 iterations (first 5000 discarded), n.thin =
10
```

```
##  n.sims = 13500 iterations saved
##           mean    sd  2.5%   25%   50%   75% 97.5%  Rhat n.eff
## meanp1   0.225 0.040 0.154 0.198 0.223 0.251 0.307 1.001  8700
## meanp2   0.419 0.085 0.259 0.360 0.418 0.476 0.591 1.001 14000
## meanp3   0.396 0.058 0.285 0.356 0.396 0.436 0.510 1.001 14000
## meanp2p1 1.921 0.532 1.052 1.545 1.865 2.234 3.119 1.001 14000
## meanp3p1 1.822 0.454 1.100 1.501 1.771 2.084 2.858 1.001 14000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence,
Rhat=1).

media2 <- resultado2$mean
media2

## $meanp1
## [1] 0.2252904
##
## $meanp2
## [1] 0.4192538
##
## $meanp3
## [1] 0.3964614
##
## $meanp2p1
## [1] 1.921332
##
## $meanp3p1
## [1] 1.821628

p_quantiles <- data.frame(
  Variable = c("meanp1", "meanp2", "meanp3", "meanpp2p1", "meanp3p1"),
  per2.5 = round(c(quantile(attach_res2$meanp1, probs = c(0.025,
0.975))[1],
            quantile(attach_res2$meanp2, probs = c(0.025, 0.975))[1],
            quantile(attach_res2$meanp3, probs = c(0.025, 0.975))[1],
            quantile(attach_res2$meanp2p1, probs = c(0.025, 0.975))[1],
            quantile(attach_res2$meanp3p1, probs = c(0.025,
0.975))[1]),4),
  per97.5 = round(c(quantile(attach_res2$meanp1, probs = c(0.025,
0.975))[2],
            quantile(attach_res2$meanp2, probs = c(0.025, 0.975))[2],
            quantile(attach_res2$meanp3, probs = c(0.025, 0.975))[2],
            quantile(attach_res2$meanp2p1, probs = c(0.025, 0.975))[2],
            quantile(attach_res2$meanp3p1, probs = c(0.025,
0.975))[2]),4)
)

# Mostrar los resultados en dos dataframe
print(p_quantiles)
```

```
##    Variable per2.5 per97.5
## 1    meanp1 0.1538  0.3072
## 2    meanp2 0.2587  0.5915
## 3    meanp3 0.2850  0.5096
## 4 meanpp2p1 1.0520  3.1187
## 5  meanp3p1 1.1000  2.8580
```

Interpretación

En el modelo alt, π1 representa la probabilidad de bajo peso al nacer para la raza de referencia (por ejemplo, raza blanca). El valor de $meanp1 sugiere que, en promedio, la probabilidad de bajo peso al nacer para esta raza de referencia es aproximadamente del 22.5%.

En el modelo alt, π2 representa la probabilidad de bajo peso al nacer para la raza específica 2 (raza negra). El valor de $meanp2 sugiere que, en promedio, la probabilidad de bajo peso al nacer para esta raza es aproximadamente del 41.9%.

En el modelo alt, π3 representa la probabilidad de bajo peso al nacer para la raza específica 3. El valor de $meanp3 sugiere que, en promedio, la probabilidad de bajo peso al nacer para esta raza específica es aproximadamente del 39.6%.

Parámetro cociente π2/π1. En este contexto, significa que, en promedio, la probabilidad de bajo peso al nacer para la raza específica 2 (raza negra) es aproximadamente 1.92 veces mayor que la probabilidad para la raza de referencia.

Este valor es el cociente π3/π1. En este contexto, significa que, en promedio, la probabilidad de bajo peso al nacer para la raza específica 3 es aproximadamente 1.82 veces mayor que la probabilidad para la raza de referencia.

De hecho, si nos fijamos en los IC95 vemos que son bastante estrechos por lo que existe gran precisión en la estimación del parámetro. Por tanto, matizamos que existe aproximadamente el doble de riesgo relativo de tener un hijo de bajo peso si la madre es de raza negra o de otra raza distinta a raza blanca.

Conclusión

En este caso, hemos realizado un analisis de inferencia bayesiana, que en comparación con el modelo clásico, nos permite utilizar distribuciones a priori y la distribución posterior para estimar los parámetros, proporciona intervalos de credibilidad en vez de confianza (más complejos de definir su concepto) y proporciona distribuciones de probabilidad directas para los parámetros.

Todo ello, facilita el manejo de pequeñas muestras y datos desbalanceados, permite la incorporación natural de información previa y produce mayor flexibilidad en el modelado haciendo a este método de gran utilidad. Sin embargo, la elección entre métodos depende del problema específico, las metas de inferencia y las características de los datos.