# Análisis Bayesiano de Regresión Logística y Modelos Jerárquicos

*Eduardo García Tapia*

## Ejercicio 1.

*Se realizó un pequeño experimento con el fin de medir el riesgo de cierto tipo de tumor en un grupo de ratas, dadas diferentes dosis de una droga. El propósito del estudio es estudiar la relación entre la dosis y la respuesta. i.e. la tasa a la que el riesgo de tumor crece o decrece como función de la dosis. Los datos del experimento se presentan en la Tabla 1, donde $x$ representa el nivel de la dosis, mientras que $n_*$ y $y_*$ denotan, respectivamente, el número de ratas tratadas y el número de ratas que presentan tumor en cada nivel $(x = 0, 1, 2)$*

*Tabla 1*

| $x$ | $n_x$ | $y_x$ |
|---|---|---|
| *0* | *14* | *4* |
| *1* | *34* | *4* |
| *2* | *43* | *2* |

*Sea $\pi_x$ la probabilidad de que una rata en el grupo $x$ desarrolle un tomor. COnsidera el modelo*

$$Y_x \sim Bin(\pi_x, n_x) \quad ; \quad (x = 0, 1, 2, )$$

*Dado que las investigadoras están interesadas en la forma como varía $\pi_x$ en función de la dosis $x$, propusieron el modelo:*

$$logit(\pi_x) = \alpha + \beta_x \quad ; \quad (x = 0, 1, 2)$$

*El parámetro de interés para las investigadoras es la pendiente $\beta$, pero no cuentan con información inicial sobre su valor.*

*Proporcionen un resumen (lo más completo posible) de la distribución final de $\beta$ suponiendo una distribución inicial no informativa en la $\alpha$ y $\beta$ se asumen independientes, con $\alpha \sim N(0, 1000)$ y $\beta \sim N(0, 1000)$; esto es, con media 0 y varianza 1000.*

### 1.1) Definir datos observados.

```
# Librerías necesarias
library(foreign)
library(lattice)
library(R2jags)

# Semilla para generar números aleatorios
set.seed(314159)

# Datos observados en el experimento
dosis = c(0, 1, 2)
n = c(13, 34, 34)
y = c(4, 4, 2)

# Valores x para curva de regresión
dosis_seq = seq(0, 2, length.out=50)

# Guardar datos muestrales en una sola variable
datos_1 = list("dosis"=dosis, "n"=n, "y"=y)
```

**1.2) Utilizar JAGS para el análisis bayesiano.**

```
# Definir los valores de interés
valores_interes  =  c("alpha", "beta")

# Definir valores iniciales
iniciales_1 = list("alpha"=-1.5, "beta"=-1.5)
iniciales_2 = list("alpha"=-0.5, "beta"=-0.5)
valores_iniciales = list(iniciales_1, iniciales_2)

# Definir el modelo probabilístico
bugs_1 = function()  {

  # Distribución de la respuesta y el parámetro theta
  for(i in 1:3){
    logit(p[i]) = alpha + beta*dosis[i]
    y[i] ~ dbin(p[i],n[i]) }

  # Distribución de los coeficientes de regreción
  beta ~ dnorm(0, 0.001)
  alpha ~ dnorm(0, 0.001) }

# Se utiliza para no mostrar salida de texto en el PDF
salida_texto =  capture.output({

# Correr modelo JAGS
modelo_1 = jags(model.file=bugs_1, parameters.to.save=valores_interes,
                data=datos_1, n.chains=2, n.thin=100, DIC=FALSE,
                n.burnin=50000, n.iter=200000, inits=valores_iniciales) })

# Convertir a objeto MCMC
mcmc_1  =  as.mcmc(modelo_1)
```

**1.3) Análisis de convergencia.**

Para analizar la convergencia y valides del muestreo Gibbs que se realizó, primero se revisan los parámetros *R-hat* y el número efectivo de simulaciones para los parámetros. En la salida del modelo se observan valores de $R - hat$ cercanos a 1 y el nñúmero efectivo de simulaciones es alto. Por lo tanto, en este paso se admite la convergencia de la cadena de Markov.

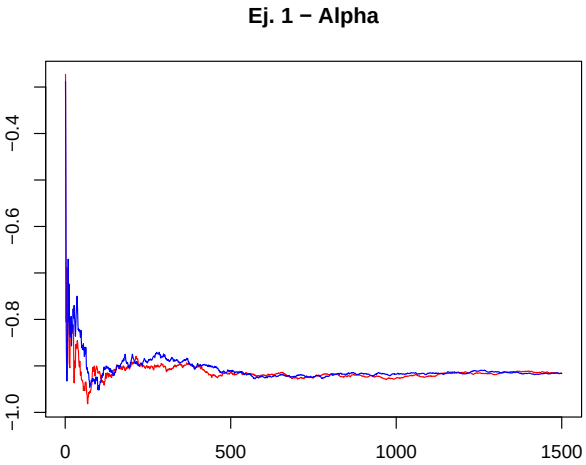
Convergenca Ejercicio 1

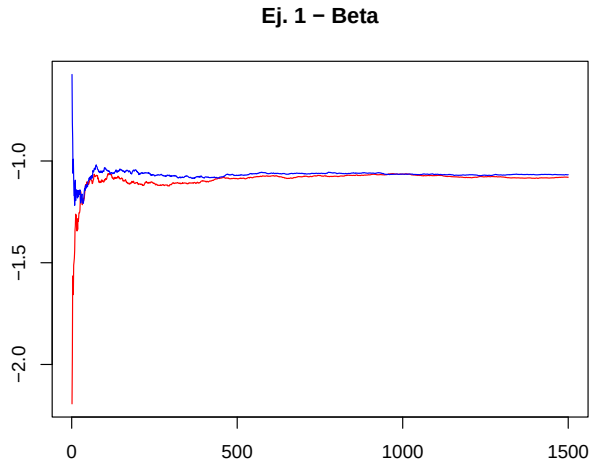
| Parámetro | $R_{hat}$ | $n.eff$ |
|---|---|---|
| $\alpha$ | 1.001 | 3000 |
| $\beta$ | 1.001 | 3000 |

Como segundo análisis de convergencia se observan los promedios ergódicos para las simulaciones en cada una de las cadena. Lo esperado es que los valores convergen, por lo que acepta la simulación obtenida.

```
# Número de muestras en la cadena
M = length(mcmc_1[[1]][,"alpha"])

# Cuadrícula para graficar
par(mfrow=c(1, 2))

# Gráfica para alpha
```

```r
prom_erg_1 = cumsum(mcmc_1[[1]][,"alpha"]) / (1:M)
prom_erg_2 = cumsum(mcmc_1[[2]][,"alpha"]) / (1:M)
y_max = max(prom_erg_1, prom_erg_2)
y_min = min(prom_erg_1, prom_erg_2)
plot(1:M, prom_erg_1, main="Ej. 1 - Alpha",type="l", xlab="", ylab="", ylim=c(y_min, y_max), col="red")
lines(1:M, prom_erg_2, col="blue")

# Gráfica para beta
prom_erg_1 = cumsum(mcmc_1[[1]][,"beta"]) / (1:M)
prom_erg_2 = cumsum(mcmc_1[[2]][,"beta"]) / (1:M)
y_max = max(prom_erg_1, prom_erg_2)
y_min = min(prom_erg_1, prom_erg_2)
plot(1:M, prom_erg_1, main="Ej. 1 - Beta",type="l", xlab="", ylab="", ylim=c(y_min, y_max), col="red")
lines(1:M, prom_erg_2, col="blue")
```




**1.4) Resumen de la información.**

## Ejercicio 2.

*Dado que el tamaño de las muestras en el problema anterior ex muy pequeño, y en vista de la falta de información inicial, las investigadoras se dieron a la tarea de buscar información relevante en la literatura. Como producto de esta labor encontraron datos de 10 estudios similares con ratas de la misma cepa. Desfortunadamente, todos estos datos correspondían a controles: es decir, ratas a las que no les aplicó la droga. Los datos se presentan en la Tabla 2a. Aquí $n_{0,i}$ y $y_{0,i}$ denota, respectivamente, el número total de ratas y el número de ratas que presentaron un tumor en el í-ésimo estudio* $(1 = 1, 2, ..., 10)$

*Tabla 2a*

| $x$ | $n_x$ | $y_x$ |
|---|---|---|
| *1* | *10* | *1* |
| *2* | *13* | *2* |
| *3* | *48* | *10* |
| *4* | *19* | *5* |
| *5* | *29* | *0* |
| *6* | *18* | *0* |
| *7* | *25* | *2* |
| *8* | *49* | *5* |
| *9* | *48* | *9* |
| *10* | *19* | *4* |

No satisfechas con estos datos, las investigadoras siguieron buscando trabajos recientes (no publicados). Finalmente encontraron dos reportes muy relevantes, de los cuales extrajeron estos datos.

*Tabla 2b*

| $x$ | $n_x$ | $y_x$ |
|---|---|---|
| *0* | *7* | *3* |
| *1* | *16* | *5* |
| *2* | *18* | *2* |

*Tabla 2c*

| $x$ | $n_x$ | $y_x$ |
|---|---|---|
| *0* | *15* | *2* |
| *1* | *11* | *1* |
| *2* | *9* | *0* |

*En vista de que los datos en la Tabla 2a solo se recabó información de controles, el nivel de la dosis es* $x = 0$ *en todos esos casos. Por lo tanto el modelo que propusieron para esos datos es:*

$$.Y_i \sim Bin(\pi_{0,i}, n_{0,i}) \quad ; \quad (x = 0) \; ; \; (i = 1, 2, ..., 10)$$

*donde:*

$$logit(\pi_{0,i}) = \alpha_i \quad ; \quad (i = 1, 2, ..., 10)$$

*Por otra parte, para los datos de las Tablas 2b y 2c (estudios 11 y 12), las investigadoras supusieron un modelo de la misma forma que el del Problema 1, es decir:*

$$.Y_{x,i} \sim Bin(\pi_{x,i}, n_{x,i}) \quad ; \quad (x = 0, 1, 2) \; ; \; (i = 11, 12)$$

*con:*

$$.logit(\pi_{x,i} = \alpha_i + \beta_i x) \quad ; \quad (x = 0, 1, 2) \; ; \; (i = 11, 12)$$

*Para simplificar el análisis en esta etapa, las investigadoras decidieron considerar todos estos estudios suficientemente similares como para suponer que los datos de las Tablas 1, 2a, 2b y 2c provienen de un solo experimento de manera que* $\alpha_1 = \alpha_2 = ... = \alpha_{12} = \alpha$ *y* $\beta_1 = ... = \beta_{11} = \beta_{12} = \beta$

*Utilizando la misma distribución inicial que en el Ejercicio 1, proporcionen un resumen (lo más completo posible) de la distribución final de $\beta$.*

**2.1) Definir datos observados.**

```
# Datos observados en los experimentos
n = c(295, 61, 61)
y = c(47, 10, 4)

# Guardar datos muestrales en una sola variable
datos_2 = list("dosis"=dosis, "n"=n, "y"=y)
```

**2.2) Utilizar JAGS para el análisis bayesiano.**

```
# Definir los valores de interés
valores_interes  =  c("alpha", "beta")

# Definir valores iniciales
iniciales_1 = list("alpha"=-2.0, "beta"=-2.0)
iniciales_2 = list("alpha"=-0.0, "beta"=-0.0)
valores_iniciales = list(iniciales_1, iniciales_2)

# Definir el modelo probabilístico
bugs_2 = function()  {

  # Distribución de la respuesta y el parámetro theta
  for(i in 1:3){
    logit(p[i]) = alpha + beta*dosis[i]
    y[i] ~ dbin(p[i],n[i]) }

  # Distribución de los coeficientes de regreción
  beta ~ dnorm(0, 0.001)
  alpha ~ dnorm(0, 0.001) }

# Se utiliza para no mostrar salida de texto en el PDF
salida_texto =  capture.output({

# Correr modelo JAGS
modelo_2 = jags(model.file=bugs_2, parameters.to.save=valores_interes,
                data=datos_2, n.chains=2, n.thin=100, DIC=FALSE,
                n.burnin=50000, n.iter=200000, inits=valores_iniciales) })

# Convertir a objeto MCMC
mcmc_2  =  as.mcmc(modelo_2)
```

**2.3) Análisis de convergencia.**

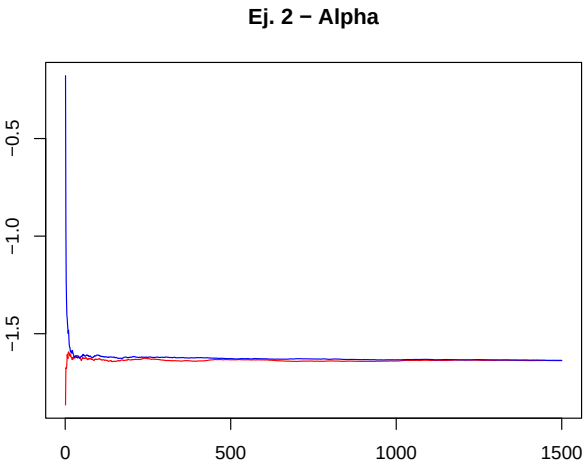
Para el análisis de convergencia del modelo 2 se vuelven a utilizar las métricas *Rhat* y *n.eff*. En la salida debajo se observa que, efectivamente, se tienen convergencia en el muestreo Gibbs.

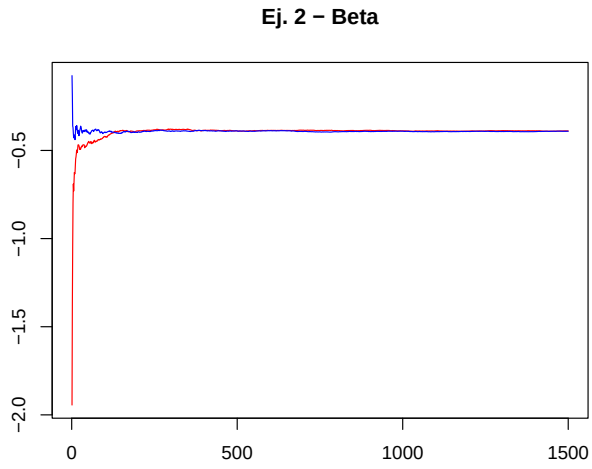
Convergenca Ejercicio 2

| Parámetro | $R_{hat}$ | $n.eff$ |
|---|---|---|
| $\alpha$ | 1.001 | 3000 |
| $\beta$ | 1.001 | 3000 |

La segund etapa del análisis se basa en revisar los promedios ergódicos. Se observa que existe convergencia para la simulación de dos cadenas. Por lo tanto, se acepta el modelo como válido, pues existe convergencia.

```r
# Número de muestras en la cadena
M = length(mcmc_2[[1]][,"alpha"])

# Cuadrícula para graficar
par(mfrow=c(1, 2))

# Gráfica para alpha
prom_erg_1 = cumsum(mcmc_2[[1]][,"alpha"]) / (1:M)
prom_erg_2 = cumsum(mcmc_2[[2]][,"alpha"]) / (1:M)
y_max = max(prom_erg_1, prom_erg_2)
y_min = min(prom_erg_1, prom_erg_2)
plot(1:M, prom_erg_1, main="Ej. 2 - Alpha",type="l", xlab="", ylab="", ylim=c(y_min, y_max), col="red")
lines(1:M, prom_erg_2, col="blue")

# Gráfica para beta
prom_erg_1 = cumsum(mcmc_2[[1]][,"beta"]) / (1:M)
prom_erg_2 = cumsum(mcmc_2[[2]][,"beta"]) / (1:M)
y_max = max(prom_erg_1, prom_erg_2)
y_min = min(prom_erg_1, prom_erg_2)
plot(1:M, prom_erg_1, main="Ej. 2 - Beta",type="l", xlab="", ylab="", ylim=c(y_min, y_max), col="red")
lines(1:M, prom_erg_2, col="blue")
```




**2.4. Resumen de la información.**

## Ejercicio 3

*Poco tiempo después, una de las investigadoras tuvo la oportunidad de asisistir a un curso de Análisis Bayesiano de Modelos Jerárquicos y convenció al resto del equipo de que esa es la manera más apropiada de analizar los datos con los que contaban. Específicamente, dado que todos los estudios eran similares, consideraron que podían utilizar los 12 estudios que encontraron en la literatura para complementar la información de su experimente original.*

*Las investigadoras supusieron entonces que los parámetros* $\{\alpha, \alpha_1, ..., \alpha_{12}\}$ *eran intercambiavles, con distribución común* $N(\alpha^*, \sigma_\alpha^2)$*, y también que los parámetros* $\{\beta, \beta_{11}, \beta_{12}\}$ *eran intercambiables, con distribución comñun* $N(\beta^*, \sigma_\beta^2)$*. Finalmente, tanto para* $\alpha^*$ *como para* $\beta^*$ *supusieron un distribución* $N(0, 100)$*, mientras que para* $\tau_\alpha = 1/\sigma_\alpha^2$ *y* $\tau_\beta = 1/\sigma_\beta^2$ *considerando una distribución* $Gamma(0.01, 0.01)$

*Proporcionen un resumen (lo más completo posible) de la distribución final de* $\beta$ *(la correspondiente al ejercicio 1 bajo estas condiciones.*

### 3.1) Definir datos observados.

```
# Definir datos de todos los experimentos
n = c(14,34,34,  10,13,48,19,20, 18,25,49,48,19,  7,16,18,  5,11,9)
y = c(4,4,2,  1,2,10,5,0, 0,2,5,9,4,  3,5,2,  2,1,0)

# Guardar datos muestrales en una sola variable
datos_3 = list("y"=y, "n"=n, "dosis"=dosis)
```

### 3.2 Utilizar JAGS para el análisis bayesiano.

```
# Definir los valores de interés
valores_interes  =  c("alpha[1]", "beta[1]")

# Definir valores iniciales
iniciales_1 = list("alpha[1]"=-1.5, "beta[1]"=-1.5)
iniciales_2 = list("alpha[1]"=-0.5, "beta[1]"=-0.5)
valores_iniciales = list(iniciales_1, iniciales_2)

# Definir valores iniciales
iniciales_1 = list("alpha[1]"=-1.5, "beta[1]"=-1.5)
iniciales_2 = list("alpha[1]"=-0.5, "beta[1]"=-0.5)
valores_iniciales = list(iniciales_1, iniciales_2)

# Definir el modelo probabilístico
bugs_3 = function()  {



  # Nivel 1 --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
  # Experimentos con dosis
  for(i in 1:3){

    # Tabla 1
    logit(p[i]) = alpha[1] + beta[1]*dosis[i]
    y[i] ~ dbin(p[i], n[i])

    # Tabla 2b
    logit(p[i+13]) = alpha[12] + beta[2]*dosis[i]
    y[i+13] ~ dbin(p[i+13], n[i+13])
```

```
    # Tabla 2c
    logit(p[i+16]) = alpha[13] + beta[3]*dosis[i]
    y[i+16] ~ dbin(p[i+16], n[i+16]) }

  # Experimento de controles
  for(i in 4:13){

    # Tabla 2a
    logit(p[i]) = alpha[i-2]
    y[i] ~ dbin(p[i], n[i]) }



  # Nivel 2 --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
  # Distribución de los coeficientes de regreción
  for(i in 1:13){
    alpha[i] ~ dnorm(alpha_media, alpha_precision) }

  for(i in 1:3){
    beta[i] ~ dnorm(beta_media, beta_precision) }



  # Nivel 3 --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
  alpha_media ~ dnorm(0, 0.001)
  beta_media ~ dnorm(0, 0.001)
  alpha_precision ~ dgamma(0.01,0.01)
  beta_precision ~ dgamma(0.01,0.01) }

# Se utiliza para no mostrar salida de texto en el PDF
salida_texto =  capture.output({

# Correr modelo JAGS
modelo_3 = jags(model.file=bugs_3, parameters.to.save=valores_interes,
                data=datos_3, n.chains=2, n.thin=50, DIC=FALSE,
                n.burnin=50000, n.iter=300000, inits=valores_iniciales) })

# Convertir a objeto MCMC
mcmc_3  =  as.mcmc(modelo_3)
```

### 3.3) Análisis de convergencia.

Para el análisis de convergencia del modelo 3 se vuelven a utilizar las métricas *Rhat* y *n.eff*. En la salida debajo se observa que, efectivamente, se tienen convergencia en el muestreo Gibbs.

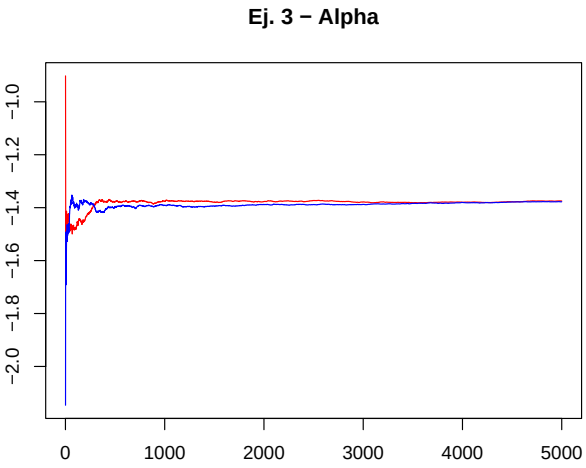
Convergenca Ejercicio 3

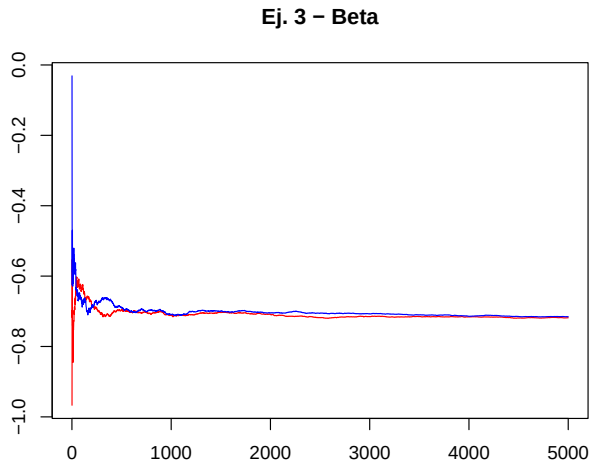
| Parámetro | $R_{hat}$ | $n.eff$ |
|---|---|---|
| $\alpha$ | 1.001 | 10,000 |
| $\beta$ | 1.001 | 10,000 |

La segund etapa del análisis se basa en revisar los promedios ergódicos. Se observa que existe convergencia para la simulación de dos cadenas. Por lo tanto, se acepta el modelo como válido, pues existe convergencia.

```
# Número de muestras en la cadena
M = length(mcmc_3[[1]][,"alpha[1]"])
```

```r
# Cuadrícula para graficar
par(mfrow=c(1, 2))

# Gráfica para alpha
prom_erg_1 = cumsum(mcmc_3[[1]][,"alpha[1]"]) / (1:M)
prom_erg_2 = cumsum(mcmc_3[[2]][,"alpha[1]"]) / (1:M)
y_max = max(prom_erg_1, prom_erg_2)
y_min = min(prom_erg_1, prom_erg_2)
plot(1:M, prom_erg_1, main="Ej. 3 - Alpha",type="l", xlab="", ylab="", ylim=c(y_min, y_max), col="red")
lines(1:M, prom_erg_2, col="blue")

# Gráfica para beta
prom_erg_1 = cumsum(mcmc_3[[1]][,"beta[1]"]) / (1:M)
prom_erg_2 = cumsum(mcmc_3[[2]][,"beta[1]"]) / (1:M)
y_max = max(prom_erg_1, prom_erg_2)
y_min = min(prom_erg_1, prom_erg_2)
plot(1:M, prom_erg_1, main="Ej. 3 - Beta",type="l", xlab="", ylab="", ylim=c(y_min, y_max), col="red")
lines(1:M, prom_erg_2, col="blue")
```




### 3.4) Resumen de la información.

```r
# Número de iteraciones para calcular valor de pi en la distr. binomial
M = length(mcmc_3[[1]][,"alpha[1]"])
k = length(dosis_seq)

# Usados para guardar cuantiles del calculo de pi-binomial
p025 = c()
p500 = c()
p975 = c()

# Calcular pi-binomial para cada alpha y beta simulada
for( i in 1:k){
  pi_binomial = c()
  for(j in 1:M){
    eta = exp(mcmc_3[[1]][,"alpha[1]"][j] + dosis_seq[i]*mcmc_3[[1]][,"beta[1]"][j])
    pi_binomial = c(pi_binomial, eta / (1+eta)) }

  # Guardar cuantiles de cálculo de pi-bin
  p025 = c(p025, quantile(pi_binomial, 0.025))
```

```r
  p500 = c(p500, quantile(pi_binomial, 0.500))
  p975 = c(p975, quantile(pi_binomial, 0.975)) }

# Generar gráfica
y_max = max(p975)
y_min = min(p025)
plot(dosis_seq, p025, type="l", ylim=c(y_min, y_max), xlab="Dosis", ylab="Probabilidad de tumor", main='
lines(dosis_seq, p500)
lines(dosis_seq, p975)
```