Inferencia Bayesiana

Alumno:
Huertas Quispe, Anthony Enrique ***Cod:*** *20173728*

Semestre: *2018-I*

Tema: PC 3

PROF. VICTOR GIANCARLO SAL Y ROSAS

Pontificia Universidad Católica del Perú
Escuela de Posgrado
Maestría en Estadística

Problema 1. (Distribución Rayleigh)

La distribución Rayleigh, denotada por $X \sim Rayleigh)\sigma$, es usada para modelar el tiempo de vida de un sujeto donde la función de riesgo crece de manera lineal.

$$f(x) \quad = \quad \frac{x}{\sigma^2} \exp\left[-\left(\frac{x^2}{2\sigma^2}\right)\right]$$

para $x \in [0, \infty]$.

Listing 1: Función de distribución Rayleigh

```
f <- function(x, sigma=2) { (x/sigma^2)*exp(-x^2/(2*sigma^2))}
```

Asumiendo que $\sigma = 2$, responda las siguiente preguntas

a) Use el algoritmo Metropolis - Hasting para generar observaciones de esta distribución. Use como función generadora

$$x^* \sim J(\dot{|}x^{(s-1)}) = dchisq(df = x^{(s-1)})$$

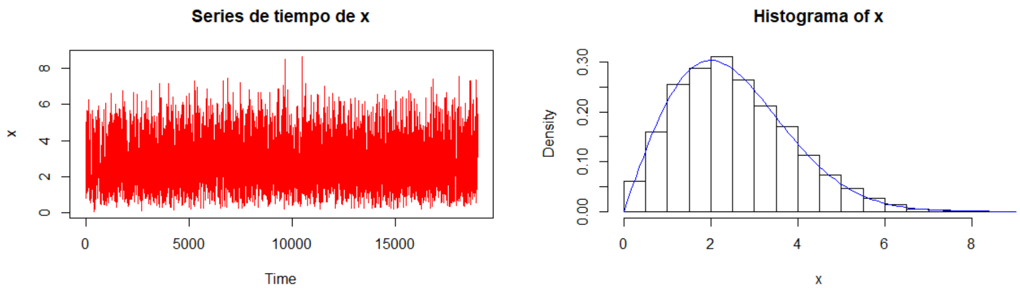
Corrobore que puede recuperar la distribución usando la función **drayleigh** de la librería **VGAM**

Solución.

Listing 2: Metropolis-hasting / Función Generadora: χ^2

```
N <- 20000
x <- numeric(N)

# Valor inicial
x[1] <- 0.5

# Algoritmo Metropolis-Hasting
for(i in 2:N){
        # Valor anterior
        x.start<-x[i-1]

        # Proponer (Funcion generadora : chi^2)
        x.prop<-rchisq(1,df=x.start) # valor candidato

        # Regla de aceptacion
        u<-runif(1)
        px.start_x.prop <- dchisq(x.start,df=x.prop)
        px.prop_x.start <- dchisq(x.prop,df=x.start)
        r<-min(1,(f(x.prop)*px.start_x.prop)/(f(x.start)*px.prop_x.start))
        if(u < r){x[i]<-x.prop}else{x[i]<-x.start} #acepta al candidato
}

# Valores simulados (Eliminando primeras 1000 observaciones)
x <- x[-c(1:1000)]
acf(x)
ts.plot(x,col=2,main=paste("Series de tiempo de x"))

# Histograma de distribucion
hist(x,prob=T,main=paste("Histograma of x"))
curve(drayleigh(x, scale=2), add=TRUE, col="blue")
```

□



(a) Series de tiempo (b) Histograma

Figura 1: Simulación de una distribucion Rayleigh por M-H usando como función generadora una $\chi^2_{(k)}$.

b) La distribución normal truncada en $[a, b]$, es denotada por $X \sim NT(\mu, \sigma^2, a, b)$ y tiene como función de densidad

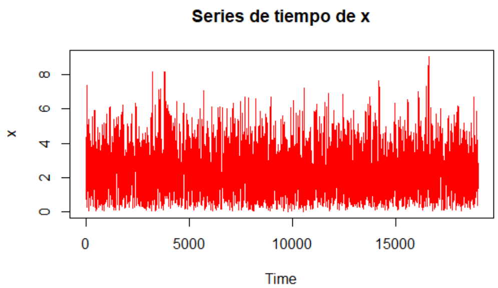
$$f(x) = \frac{\phi\left(\dfrac{x-\mu}{\sigma}\right)}{\sigma\left[\Phi\left(\dfrac{b-\mu}{\sigma}\right)\Phi\left(\dfrac{a-\mu}{\sigma}\right)\right]}$$

donde ϕ y Φ son la función de densidad y la función de distribución acumulada de la normal estándar, respectivamente. Se puede generar observaciones de esta distribución usando la función **rtnorm** de la librería **msm**. Repita la pregunta a) usando una normal truncada con función generadora.

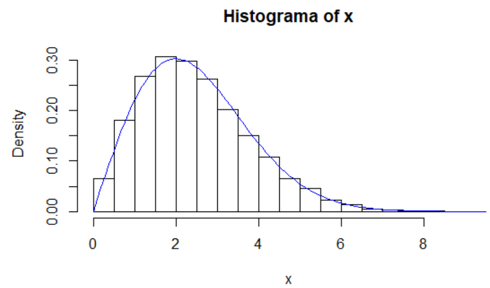
Solución.

Listing 3: Metropolis-hasting / Función Generadora: TN (Normal Truncada).

```
library(msm)
N <- 20000
x <- numeric(N)

# Valor inicial
x[1] <-0.5

# Algoritmo Metropolis-Hasting
for(i in 2:N){
        # Valor anterior
        x.start<-x[i-1]

        # Proponer (Funcion generadora : TN)
        x.prop<-rtnorm(1,mean=x.start, sd=1,lower=0, upper=Inf)

        # Regla de aceptacion
        u<-runif(1)
        px.start_x.prop <- dtnorm(x.start,mean=x.prop, sd=1,lower=0, upper=Inf)
        px.prop_x.start <- dtnorm(x.prop,mean=x.start, sd=1,lower=0, upper=Inf)
        r<-min(1,(f(x.prop)*px.start_x.prop)/(f(x.start)*px.prop_x.start))
        if(u < r){x[i]<-x.prop}else{x[i]<-x.start}
}

# Valores simulados
x <- x[-c(1:1000)]
acf(x)
ts.plot(x,col=2,main=paste("Series de tiempo de x"))

hist(x,prob=T,main=paste("Histograma of x"))
curve(drayleigh(x, scale=2), add=TRUE, col="blue")
```



(a) Series de tiempo



(b) Histograma

Figura 2: Simulación de una distribucion Rayleigh por M-H usando como función generadora una Normal Truncada.

□

c) Use una distribución Gamma como función generadora

$$J(y \mid x^{(s)}) = dgamma\left(x^*, \eta, \frac{\eta}{x^{(s-1)}}\right) = \frac{\left(\frac{\eta}{x^{(s-1)}}\right)^{\eta}}{\Gamma(\eta)} y^{\eta-1} \exp\left[-\frac{\eta}{x^{(s)}} y\right]$$

y reputa la pregunta a) considerando los valores $\eta = 1$ y $\eta = 2$.

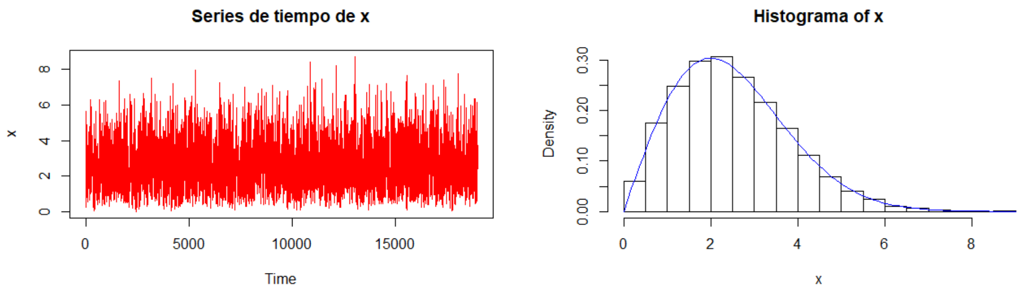
Solución.

Listing 4: Metropolis-hasting / Función Generadora: *Gamma*.

```
rayleigh_MH_gamma <- function(N=20000,sigma = 2,eta = 1){
        x <- numeric(N)

        # Valor inicial
        x[1] <-0.5
        for(i in 2:N){
                # Valor anterior
                x.start<-x[i-1]

                # Proponer
                x.prop<-rgamma(1,shape = eta, rate = eta/x.start)# valor candidato

                # Regla de aceptacion
                u<-runif(1)
                px.start_x.prop <- dgamma(x.start,shape = eta, rate = eta/x.prop)
                px.prop_x.start <- dgamma(x.prop,shape = eta, rate = eta/x.start)
                r<-min(1,(f(x.prop)*px.start_x.prop)/(f(x.start)*px.prop_x.start))
                if(u < r){x[i]<-x.prop}else{x[i]<-x.start} #acepta al candidato
        }
        x
}
```

Listing 5: Valores simulados con $\eta = 1$.

```
x <- rayleigh_MH_gamma(eta=1)[-c(1:1000)]
acf(x)
ts.plot(x,col=2,main=paste("Series de tiempo de x"))

hist(x,prob=T,main=paste("Histograma of x"))
curve(drayleigh(x, scale=2), add=TRUE, col="blue")
```



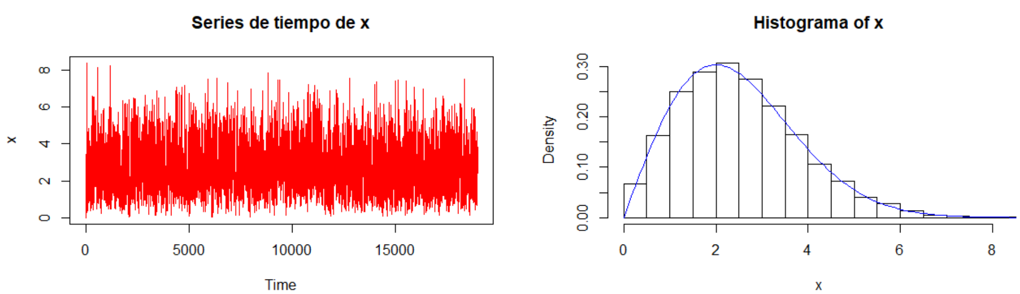
(a) Series de tiempo

(b) Histograma

Figura 3: Simulación de una distribucion Rayleigh por M-H usando como función generadora una Gamma con $\eta = 1$.

Listing 6: Valores simulados con $\eta = 2$.

```
x <- rayleigh_MH_gamma(eta=2)[-c(1:1000)]
acf(x)
ts.plot(x,col=2,main=paste("Series de tiempo de x"))

hist(x,prob=T,main=paste("Histograma of x"))
curve(drayleigh(x, scale=2), add=TRUE, col="blue")
```



(a) Series de tiempo

(b) Histograma

Figura 4: Simulación de una distribucion Rayleigh por M-H usando como función generadora una Gamma con $\eta = 2$.

□

Problema 2. (Regresión Logística Simple)

Consideremos implementar una regresión logística para estudiar los factores que podrian estar asociados con la probabilidad de caer en morosidad. En particular, deseamos estudiar la relación entre la **edad** del cliente y la probabilidad de caer en morosidad. La base de datos se puede leer de la siguiente forma

Listing 7: Importando datos

```
credito <- read.csv(file.choose())
credito <- credito[,1:10]
names(credito) <- c("id","censor","tiempo","bancarizado","sexo","soltero","edad","pcali",
                    "tarjeta","linea")
```

a) Asumiendo que

$$\beta = (\beta_0, \beta_1) \sim N_2(\mu, \Sigma)$$

construya la distribución a posteriori de (β_0, β_1)

Solución. Considerando la variable X : *Edad* y Y : *censor* entonces

$$Y \mid X = x \sim Bernoulli(\pi(x))$$

donde

$$logit(\pi(x)) = \beta_0 + \beta_1 x = \begin{pmatrix} 1 \\ x \end{pmatrix}^T \beta = \mathbf{x}^T\beta \quad \Rightarrow \quad \pi(x) = \frac{\exp(\mathbf{x}^T\beta)}{1+\exp(\mathbf{x}^T\beta)}$$

La distribución a posteriori vendría representada, en parte, por

$$\begin{aligned}
p(\beta \mid \mathbf{Y}) &\propto p(\mathbf{Y} \mid \beta)p(\beta) \\
&\propto \left(\prod_{i=1}^{n} \left(\frac{\exp(\mathbf{x}_i^T\beta)}{1+\exp(\mathbf{x}_i^T\beta)}\right)^{y_i} \left(\frac{1}{1+\exp(\mathbf{x}_i^T\beta)}\right)^{1-y_i}\right) \times \exp\left(-\frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\right) \\
&\propto \left(\prod_{i=1}^{n} \frac{(\exp(\mathbf{x}_i^T\beta))^{y_i}}{1+\exp(\mathbf{x}_i^T\beta)}\right) \times \exp\left(-\frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\right) \\
&\propto \frac{\exp\left(\sum_{i=1}^{n} y_i\mathbf{x}_i^T\beta\right)}{\prod_{i=1}^{n}\left(1+\exp(\mathbf{x}_i^T\beta)\right)} \times \exp\left(-\frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\right) \\
&\propto \frac{\exp\left(\sum_{i=1}^{n} y_i\mathbf{x}_i^T\beta - \frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\right)}{\prod_{i=1}^{n}\left(1+\exp(\mathbf{x}_i^T\beta)\right)}
\end{aligned}$$

□

b) Implemente un algoritmo de Metropolis-Hasting para calcular la distribución a posteriori de (β_0, β_1) asumiendo una función generadora normal bivariada

$$\beta^* \sim N_2\left(\beta^{(t-1)}, \tau\Sigma\right)$$

donde Σ es la matriz de covarianza del modelo de regresión frecuentista y τ es un parámetros de precisión.

Solución.

Listing 8: Metropolis-Hasting

```
# Modelo frecuentista
model.logit.freq <- glm(censor ~ edad, family="binomial",credito)

# Algoritmo Metropolis-Hasting
Logistic.M <- function(tau = 2,
                       y = credito$censor,
                       X = model.matrix(~ edad,credito),
                       Delta = diag(dim(X)[2]),
                       beta0 = rep(0,dim(X)[2]),
                       Sbeta0 = diag(dim(X)[2]),
                       beta = matrix(rep(0,dim(X)[2]),nrow=1),
                       I=20000){
        BETA  <- NULL

        for(i in 1:I){
                # Proponer
                beta.start <- rmvnorm(1,beta,tau*Delta)

                # Asignaciones previas
                pi.start <- exp(X%*%t(beta.start))*(1 + exp(X%*%t(beta.start)))^(-1)
                pi.beta <- exp(X%*%t(beta))*(1 + exp(X%*%t(beta)))^(-1)

                log.r <- sum(dbinom(y,1,pi.start,log=T)) - sum(dbinom(y,1,pi.beta,log=T))
                         + dmvnorm(beta.start,beta0,Sbeta0,log=TRUE)
                         - dmvnorm(beta,beta0,Sbeta0,log=TRUE)

                # Aceptacion
                if(log.r > log(runif(1))) beta <- beta.start
                BETA <- rbind(BETA,beta)
                colnames(BETA) <- c("beta0","beta1")
        }

        # Estimaciones
        BETA
}

#### Evaluado con Matriz de Varianza-Covarianza del modelo frecuentista
#    tanto para la funcion generadora como para la distribucion priori
#### Se esta tomando a 1 como la precision.
logistic.1  <- Logistic.M(tau = 1,
                          Delta =  summary(model.logit.freq)$cov.unscaled,
                          Sbeta0 = summary(model.logit.freq)$cov.unscaled, I = 20000)

# Graficas de distribucion, autocorrelacion
plot(as.mcmc(logistic.1[-c(1:1000),]))
acf(as.mcmc(logistic.1[-c(1:1000),]))

# Resumen de datos
summary(as.mcmc(logistic.1[-c(1:1000),]))
```

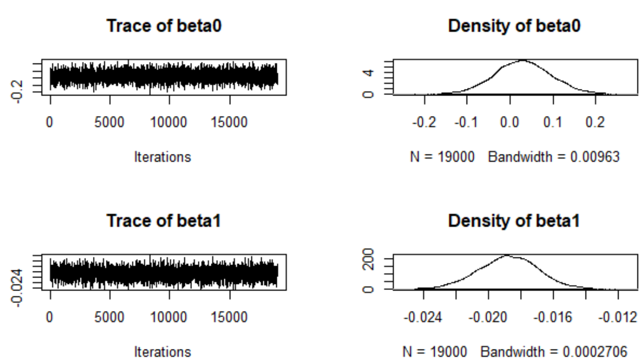


Figura 5: Gráficas de densidades

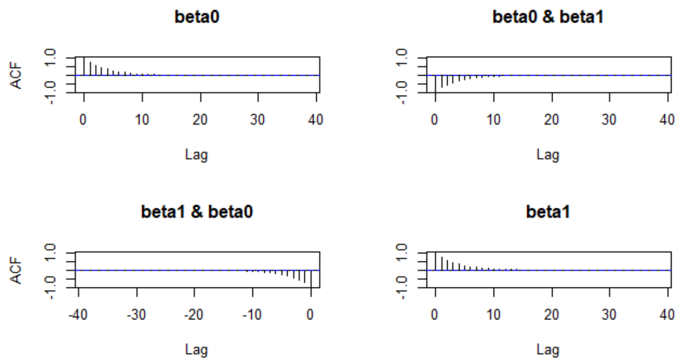


Figura 6: Gráficas de Autocorrelaciones

Se han determinado las densidades para β_0 y β_1, habiéndose eliminado los 1000 primeros valores generados con el fin de una mejor representación. Como notamos las autocorrelaciones disminuyen conforme aumenta el número de iteraciones lo cual es ideal para la estimación de los parámetros.

```
Iterations = 1:19000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 19000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean       SD  Naive SE Time-series SE
beta0  0.02963 0.065172 4.728e-04       0.001298
beta1 -0.01878 0.001832 1.329e-05       0.000036

2. Quantiles for each variable:

          2.5%      25%      50%      75%    97.5%
beta0 -0.09695 -0.01491  0.02902  0.07325  0.16027
beta1 -0.02243 -0.01999 -0.01876 -0.01753 -0.01521
```

Figura 7: Estimación de *beta*.

Se presenta un resumen de los obtenido, en donde estimaciones puntuales, en particular la media, nos dan $\beta_0 = 0.02963$ y $\beta_1 = -0.01878$.

□

c) Calcula la probabilidad de caer en morosidad a posteriori. Es decir

$$\pi(x) = \frac{\exp\left(\beta_0 + \beta_1 x\right)}{1 + \exp\left(\beta_0 + \beta_1 x\right)}$$

y proponga un ancho de banda para esta probabilidad.

Solución. El modelo tiene una forma en la cual podemos determinar la media del logaritmo de odds como

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = 0.02963 - 0.01878 \times \bar{x}$$

con $\bar{x}$ el promedio de las edades. Por lo que la estimación para la probabilidad de caer en morosidad podemos calcularla como

$$\pi(x) = \frac{\exp(0.02963 - 0.01878 \times \bar{x})}{1 + \exp(0.02963 - 0.01878 \times \bar{x})} \tag{1}$$

Ahora, para determinar el ancho de banda primero determinaremos el intervalo de confianza, al $(1-\alpha\,\%)$ para el logaritmo de odds, siendo este

$$
\begin{aligned}
IC_{logit(\pi(x))} &= \hat{\beta}_0 + \hat{\beta}_1 x \pm Z_{1-\alpha/2} \times \sqrt{Var(\hat{\beta}_0 + \hat{\beta}_1 x)} \\
&= (1,x)^T(\beta_0,\beta_1) + \pm Z_{1-\alpha/2}\sqrt{(1,x)^T Var(\hat{\beta}_0,\hat{\beta}_1)(1,x)} \\
\Rightarrow IC_{\pi(x)} &= \frac{\exp\left((1,x)^T(\beta_0,\beta_1) + \pm Z_{1-\alpha/2}\sqrt{(1,x)^T Var(\hat{\beta}_0,\hat{\beta}_1)(1,x)}\right)}{1+\exp\left((1,x)^T(\beta_0,\beta_1) + \pm Z_{1-\alpha/2}\sqrt{(1,x)^T Var(\hat{\beta}_0,\hat{\beta}_1)(1,x)}\right)}
\end{aligned}
$$

Listing 9: Metropolis-Hasting

```r
library(arm)
# beta0 y beta1 valores medios
coef = summary(as.mcmc(logistic.1[-c(1:1000),]))[1]$`statistics`[,1]

# Funcion de prediccion para x
predict.x <- function(x=0, Delta = summary(model.logit.freq)$cov.unscaled, confianza = 0.95)
        {
        tmp <- t(matrix(c(1,x))) %*% Delta %*% matrix(c(1,x))
        tmp <- qnorm(1-(1-confianza)/2)*sqrt(tmp)
        pi_x <- invlogit(t(matrix(c(1,x)))%*%coef)
        pi_x.minus <- invlogit(t(matrix(c(1,x)))%*%coef - tmp)
        pi_x.plus <- invlogit(t(matrix(c(1,x)))%*%coef + tmp)
        return(c(pi_x.minus,pi_x,pi_x.plus))
}

# Graficando ancho de banda "Edad" vs "Probabilidad de caer en morosidad"
x_raw <- seq(19,72,0.1)
N_raw <- dim(data.frame(x_raw))[1]
pi_x_raw <- rep(0,N_raw)
ht <-seq(19,72,length.out = dim(X)[1])
p_h <- matrix(0,nrow=dim(X)[1],ncol=3)

for (i in 1:N_raw) { pi_x_raw[i] = predict.x(x=x_raw[i])[2]}
for (i in 1:dim(X)[1]){ p_h[i,] <- predict.x(x = ht[i])}

plot(x = x_raw, y = pi_x_raw ,type = 'l')
polygon(c(ht,rev(ht)),c(p_h[,1],rev(p_h[,3])),col = rgb(1, 0, 0,0.5), border = NA)
lines(x= ht, y = p_h[,3], lty = 'dashed', col = 'red')
lines(x= ht, y = p_h[,2])
lines(x= ht, y = p_h[,1], lty = 'dashed', col = 'red')
```

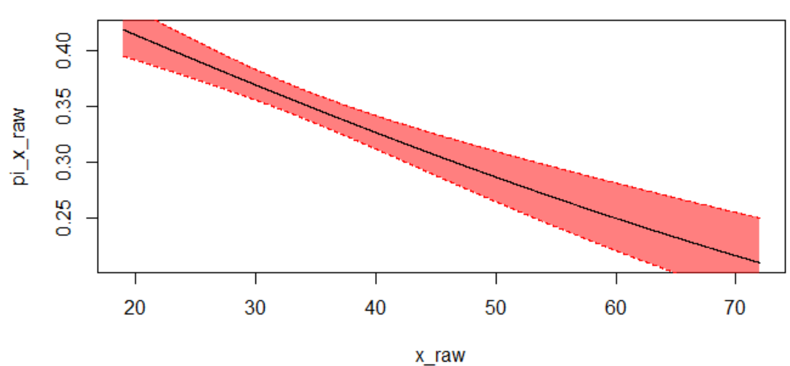


Figura 8: Banda de confianza al 95 %.

Calculamos la ecuación determinada en (1),

Listing 10: Probalidad de caer en morosidad

```
predict.x(x=mean(X[,2]))
```

Como resultado se obtiene que la probabilidad de caer en morosidad es de 0.3475226, con un intervalo de confianza de

$$[0.3346837, 0.3605872]$$

$\square$

d) Calcula la probabilidad de caer en morosidad para una nueva persona de 30 años de edad y su intervalo de credibilidad asociado

Solución. Usando la función de predicción determinada en el item c),

Listing 11: Probalidad de caer en morosidad para x=30

```
predict.x(x=30))
```

Como resulado se obtiene una probabilidad de 0.3696386, con un intervalo de confianza al 95 %

$$[0.3560335, 0.3834541]$$

$\square$

Problema 4. (Modelos Lineal Mixto)

Considere la base de datos **biomechanish.csv**. Esta base se obtuvo al realizar pruebas a 10 personas un número determinado de veces. Se desea estimar el efecto de la variable motion (X) en el delay de la persona i-ésima para intento j-ésimo.
Es decir,

$$Y_{ij} \quad = \quad X_{ij}^T\beta + b_i + \varepsilon_{ij}$$

donde $\varepsilon_{ij} \sim N(0, \sigma^2)$ y $b_i \sim N(0, \sigma_b^2)$.

a) Construya un algoritmo de Gibbs para estimar la distribución a posteriori de β, σ^2 y σ_b^2.

Tenemos lo siguiente:

$$Y_{ij} \sim N(X_{ij}^T\beta, \sigma^2 + \sigma_b^2) \quad ; \quad Y_{ij} \mid b_i \sim N(X_{ij}^T\beta + b_i, \sigma^2)$$

Además,

$$\begin{aligned} Cov(Y_{ij}, Y_{ik}) &= \sigma_b \\ Cov(Y_{ij}, Y_{i'j'}) &= 0 \text{ con } j \neq j' \end{aligned}$$

Se determinará la respectiva proporción de la distribución a posteriori de los parámetros desconocidos, (reemplazando inicialmente $\phi = 1/\sigma^2$ y $\phi_b = 1/\sigma_b^2$ como las precisiones respectivas).

$$\begin{aligned} p(\beta, \phi, \phi_b, \mathbf{b} \mid \mathbf{Y}, \mathbf{X}) &\propto p(\mathbf{Y} \mid \mathbf{X}, \beta, \phi, \phi_b, \mathbf{b}) p(\mathbf{b} \mid \phi_b) \\ &\propto \prod_i \prod_j p(Y_{ij} \mid X_{ij}, \beta, \phi, b_i) \prod_i p(b_i \mid \phi_b) \end{aligned}$$

Esto continua de la siguiente forma

$$\begin{aligned} &\propto \left[\prod_i \prod_j \phi^{1/2} \exp\left\{ -\frac{1}{2}\phi \left(Y_{ij} - X_{ij}^T\beta - b_i\right)^2 \right\} \right] \times \prod_i \phi_b^{1/2} \exp\left\{ -\frac{1}{2}\phi_b b_i^2 \right\} \\ &\propto \phi^{n/2} \exp\left\{ -\frac{1}{2}\phi \sum_{i,j} \left(Y_{ij} - X_{ij}^T\beta - b_i\right)^2 \right\} \times \phi_b^{m/2} \exp\left(-\frac{1}{2}\phi_b \sum_i b_i^2 \right) \end{aligned}$$

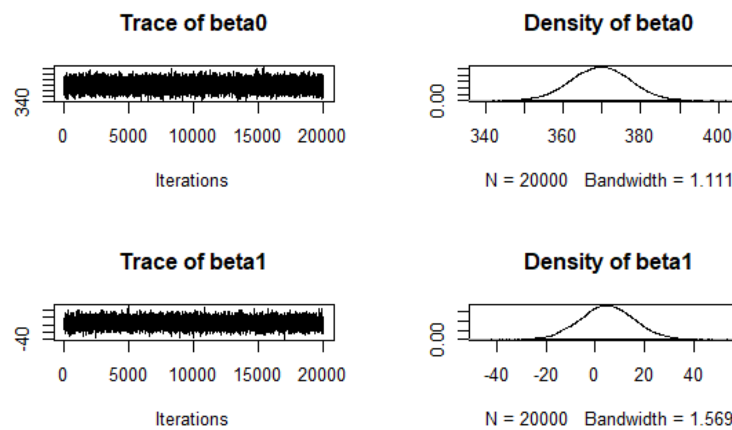
donde n es la cantidad total de observaciones y m es la cantidad de individuos.
Elaborando las distribuciones condicionales

$$\begin{aligned} b_i \mid \beta_0, \beta_1, \phi, \phi_b &\sim N\left(\frac{\phi \sum_j (Y_{ij} - X_{ij}^T\beta)}{n_i\phi + \phi_b}, (n_i\phi + \phi_b)^{-1} \right) \\ \beta \mid \phi, \phi_b, \mathbf{b} &\sim N_2\left((XX^T)^{-1}X(Y - \mathbf{b}), \frac{(XX^T)^{-1}}{\phi} \right) \\ \phi \mid \beta_0, \beta_1, \phi_b &\sim Gamma\left(\frac{n}{2} + 1, \frac{1}{2} \sum_{i,j} \left(Y_{ij} - X_{ij}^T\beta - b_i\right)^2 \right) \\ \phi_b \mid \beta_0, \beta_1, \phi &\sim Gamma\left(\frac{m}{2} + 1, \frac{1}{2} \sum_i b_i^2 \right) \end{aligned}$$

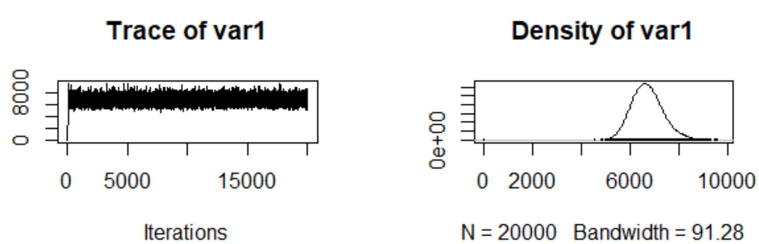
Listing 12: Probalidad de caer en morosidad

```r
library(MCMCpack)
library(data.table)
library(LearnBayes)

bio <- read.csv(file.choose())
bio <- data.frame(bio)
attach(bio)

bio <- bio[order(Subject),]
m <- 10
ni <- numeric(m)
for (i in 1:m){
        ni[i] <- dim(bio[c(bio$Subject==i),])[1]
}

database<-list()
a=c(0,ni)
for (i in 1:10){
        database[[i]] <- bio[(1+sum(a[1:i])):sum(a[1:i+1]),]
}

y <- bio$Delayms
x <- bio$Motion
X <- cbind(1,x)
n <- length(y)

# Simulaciones
I <- 20000
beta <- matrix(0,nrow = I,2)
bi <- matrix(0,nrow = I,m)
sigma2 <- rep(0,I)
sigma2.b <- rep(0,I)
phi <- rep(0,I)
phi.b <- rep(0,I)

beta[1,] <- coef(lm(y~x))
sigma2[1] <- 1
sigma2.b[1] <- 1
phi[1]<-1/sigma2[1]
phi.b[1] <- 1/sigma2.b[1]

for (h in 2:I) {
        #beta
        y_hat <- c()
        for (i in 1:m){
                y_hat <- c(y_hat,rep(bi[h-1,i],ni[i]))
        }

        mu.beta <- solve(t(X) %*% X) %*% t(X) %*% (matrix(y-y_hat))
        V.beta <- solve(t(X) %*% X)/phi[h-1]
        beta[h,] <- rmnorm(1,mu.beta,V.beta)

        #bi
        for (i in 1:m) {
                Xi <-  cbind(1,database[[i]]$Motion)
                Yi <- database[[i]]$Delayms
                mu <- phi[h-1]*sum(Yi-Xi %*% beta[h,])/(ni[i]*phi[h-1]+phi.b[h-1])
                var_bi <- 1/(ni[i]*phi[h-1]+phi.b[h-1])
                bi[h,i] <- rnorm(1,mu,sqrt(var_bi))
        }

        #phib
        phib.alpha <- m/2+1
        phib.beta <- (1/2)*sum((bi[h])^2)
        phi.b[h] <- rgamma(1,shape = phib.alpha,rate = phib.beta)
        sigma2.b[h] <- 1/phi.b[h]

        #phi
        y_hat <- c()
        for (i in 1:m){
                y_hat <- c(y_hat,rep(bi[h,i],ni[i]))
        }
```

```
        phi.alpha <- n/2 + 1
        phi.beta <- (1/2)*sum((y-y_hat-X%*%beta[h,])^2)
        phi[h] <- rgamma(1,shape = phi.alpha,rate = phi.beta)
        sigma2[h] <- 1/phi[h]
}

colnames(beta)<-c("beta0","beta1")
colnames(sigma2)<-c("sigma2")
colnames(sigma2.b)<-c("sigma2.b")

plot(as.mcmc(beta))
plot(as.mcmc(sigma2))
plot(as.mcmc(c(sigma2.b)))

summary(as.mcmc(beta))
summary(as.mcmc(sigma2))
summary(as.mcmc(sigma2.b))
```



(a) Distribución a posteriori de β



(b) Distribución a posteriori de σ^2

Figura 9: Distribución a posteriori de los parámetros desconocidos.

A continuación se presentan resumen de datos

	Media	SD
β_0	369.626	7.593
β_1	4.788	10.810
σ^2	6701.390	633.134
σ_b^2	0.0001635	0.0154166