MAT703.Deber Seminario Investigación. Ejercicios Capítulo 11 "Bayesian Data Analysis"

Fausto Fabian Crespo Fernandez

Junio 2016

0.1. Ejercicio 11.1

1. Metropolis-Hastings algorithm: Show that the stationary distribution for the Metropolis-Hastings algorithm is, in fact, the target distribution, $p(\theta|y)$.

Solución
Consideremos dos puntos θ_a y θ_b, en la iteración t etiquetadas de manera tal que $p(\theta_b|y)J_y(\theta_a|\theta_b) \geq p(\theta_a|y)J_y(\theta_b|\theta_a)$. Para mostrar que la distribución a posteriori es la distribución estacionaria de la cadena de Markov, supongamos que θ^{t-1} se escoge de la distribución a posteriori.La densidad de probabilidad incondicional de transición de θ_a a θ_b es
$p(\theta^{t-1} = \theta_a, \theta^t = \theta_b) = p(\theta_a|y)J_y(\theta_b|\theta_a)$
donde la probabilidad de aceptación es 1 debido al etiquetado a y b y la razón de razanoes de importancia es al menos 1. La densidad de probabilidad incondicional de transición de θ_b a θ_a es
$p(\theta^t = \theta_a, \theta^{t-1} = \theta_b) = p(\theta_b|y)J_y(\theta_a|\theta_b)(\frac{p(\theta_a|y)/J_y(\theta_a|\theta_b)}{p(\theta_b|y)/J_y(\theta_b|\theta_a)})$ que es la misma que la densidad de probabilidad incondicional de transición de θ_a a θ_b. Como la distribución conjunta es simétrica entonces las marginales de θ^{t-1} y θ^t son iguales y $p(\theta|y)$ es la distribución estacionaria de la cadena de Markov.Esta distribución estacionaria es única si la cadena de Markov irreducible, aperiodica y no transitoria.

0.2. Ejercicio 11.2

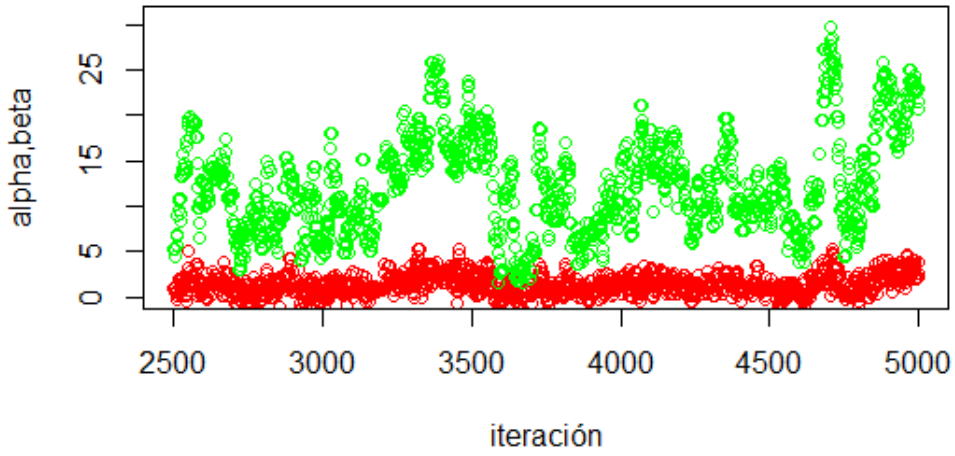
2. Metropolis algorithm: Replicate the computations for the bioassay example of Section 3.7 using the Metropolis algorithm. Be sure to define your starting points and your jumping rule. Compute with log-densities (see page 261). Run the simulations long enough for approximate convergence.

Solución
Podemos escoger la distribución de salto para los parámetros α y β como se menciona en la página 296 acerca de elegir distribuciones de salto eficientes $N(\theta^*|\theta^{t-1,c^2\Sigma})$ donde $c \approx 2{,}4/\sqrt{(d)}$ donde d es el número de parámetros a estimar, en este caso 2:α y β y Σ la matriz de varianzas de la aproximación normal a la posteriori objetivo.En R con 5000 iteraciones:
$log_posterior_update = function(alpha, beta)\{$
$t = alpha + beta * x;$
$et = exp(t);$
$z = et/(1 + et);$
$lp = sum(y * log(z) + (n - y) * log(1 - z));$

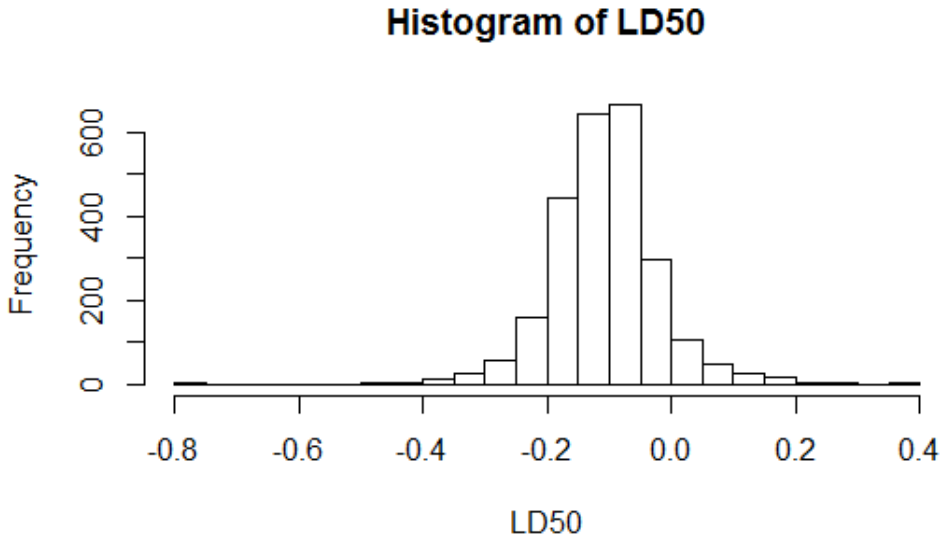
$return(lp)$
$\}$
$alpha_upd ate = function()\{$
$return(rnorm(1, alpha, 2{,}4/\sqrt{}(n.par)))$
$\}$
$beta_upd ate = function()\{$
$return(rnorm(1, beta, 2{,}4/sqrt(n.par)))$
$\}$
$x = c(-{,}863, -{,}296, -{,}053, {,}7270)$
$x = c(-{,}86, -{,}30, -{,}05, {,}73)$
$n = c(5, 5, 5, 5)$
$y = c(0, 1, 3, 5)$
$chains < -1$
$n.par = 2$
$iter < -5000$
$sims < -array(NA, c(iter, chains, n.par + 1))$
$dimnames(sims) < -list(NULL, NULL,$
$c("alpha", "beta", "log.posterior"))$
$for(iin1 : chains)\{$
$alpha = 1runif(1, -5, 10)$
$beta = 2runif(1, -10, 40)$
$log1 = log_posterior_upd ate(alpha, beta)$
$sims[1, i,] = c(alpha, beta, log1)$
$\}$
$r = 0$
$u = 0$
$for(min1 : chains)\{$
$for(tin1 : (iter - 1))\{$
$alpha < -sims[t, m, 1]$
$beta < -sims[t, m, 2]$
$alpha < -alpha_upd ate()$
$beta < -beta_upd ate()$
$log1 = log_posterior_upd ate(alpha, beta)$
$log2 = log_posterior_upd ate(sims[t, m, 1], sims[t, m, 2])$
$r = exp(log1)/(exp(log2))$
$print(r)$
$r = min(r, 1)$
$u = runif(1, 0, 1)$

$if(u < r)\{$
$sims[t+1, m,] < -c(alpha, beta, log1)$
$\}$
$else\{$
$sims[t+1, m,] < -c(sims[t, m, 1], sims[t, m, 2], log2)$
$\}$
$\}$
$\}$
$minimo = min(c(min(sims[floor(iter/2), 1, 1], min(sims[floor(iter/2) : iter, 1, 2]))))$
$maximo = max(c(max(sims[floor(iter/2) : iter, 1, 1], max(sims[floor(iter/2) :$
$iter, 1, 2]))))$
$plot(floor(iter/2) : iter, sims[floor(iter/2) : iter, 1, 1], col = "red", xlab =$
$"iteración", ylab = "alpha, beta", ylim = c(minimo - 1, maximo + 1))$
$points(floor(iter/2) : iter, sims[floor(iter/2) : iter, 1, 2], col = "green")$
$hist(sims[floor(iter/2) : iter, 1, 1], xlab = "alpha", breaks = 40, probability =$
$TRUE)$
$hist(sims[floor(iter/2) : iter, 1, 2], xlab = "beta", breaks = 40, probability =$
$TRUE)$
$LD50 = -sims[floor(iter/2) : iter, 1, 1]/(sims[floor(iter/2) : iter, 1, 2])$
$hist(LD50, xlab = "LD50", breaks = 40, probability = FALSE)$

Con lo que se obtiene con 5000 itereciones la gráficas en la segunda mitad de las iteraciones:



Donde se ve en rojo la evolución de α y en verde la evolución de β



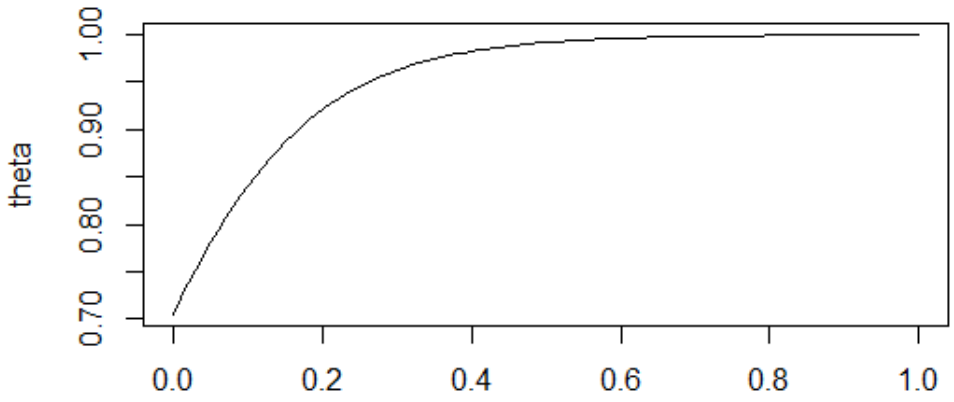
El histograma de LD50. Estos resultados son simliares a los que se obtiene con el análisis normal del 3.7. En R:
$x = c(-,863, -,296, -,053, ,7270)$

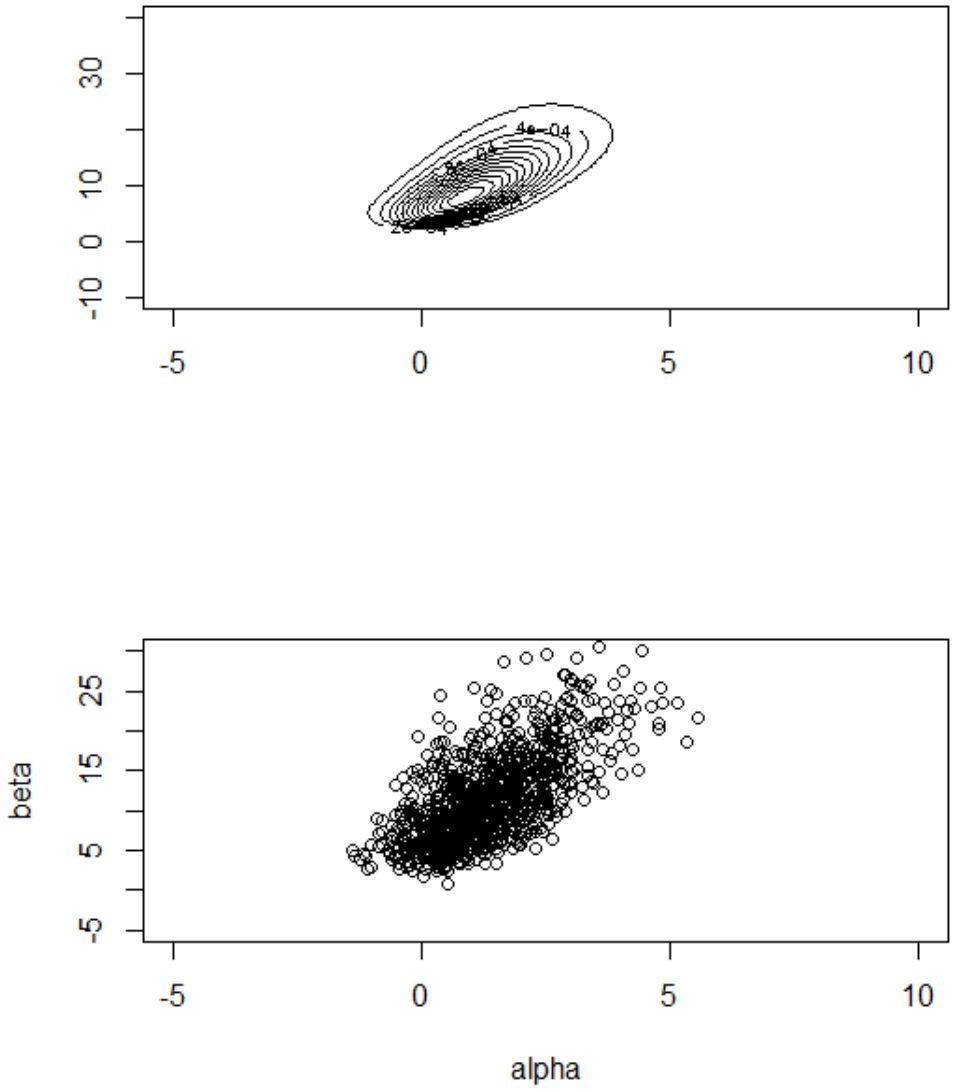
```
x = c(-,86, -,30, -,05, ,73)
n = c(5, 5, 5, 5)
y = c(0, 1, 3, 5)
response < -cbind(y, n - y)
fit < -glm(response x, family = binomial)
c = coef(fit)
curve(exp(coef(fit)[1]+coef(fit)[2]*x)/(1+exp(coef(fit)[1]+coef(fit)[2]*
x)), 0, 1, xlab = "", ylab = "theta")
covariance.matrix = summary(fit)$cov.unscaled
sd = sqrt(c(diag(covariance.matrix)))
sd
logit < -function(x){log(x/(1,0 - x))}
logitinv < -function(x){1/(1 + exp(-x))}
posteriori < -function(a, b){
tmp = 1
for(iin(1 : length(x))){
tmp2 = logitinv(a + b * x[i])
tmp = tmp * (tmp2)^y[i] * (1,0 - tmp2)^(n[i] - y[i])
}
tmp
}
densidad.posteriori < -function(a, b, y, n, x){
prod(dbinom(y, n, invlogit(a + b * x)))}
alpha = seq(-5, 10, 0,1)
beta = seq(-10, 40, 0,5)
na = length(alpha)
nb = length(beta)
z = matrix(0, na, nb)
for(iin(1 : na)){
z[i, ] = posteriori(alpha[i], beta)
}
par(mfrow = c(1, 1))
contour(alpha, beta, z, nlevels = 12)
alpha = seq(0,8, 0,95, 0,001)
beta = seq(7,6, 8,2, 0,002)
na = length(alpha)
nb = length(beta)
z = matrix(0, na, nb)
```

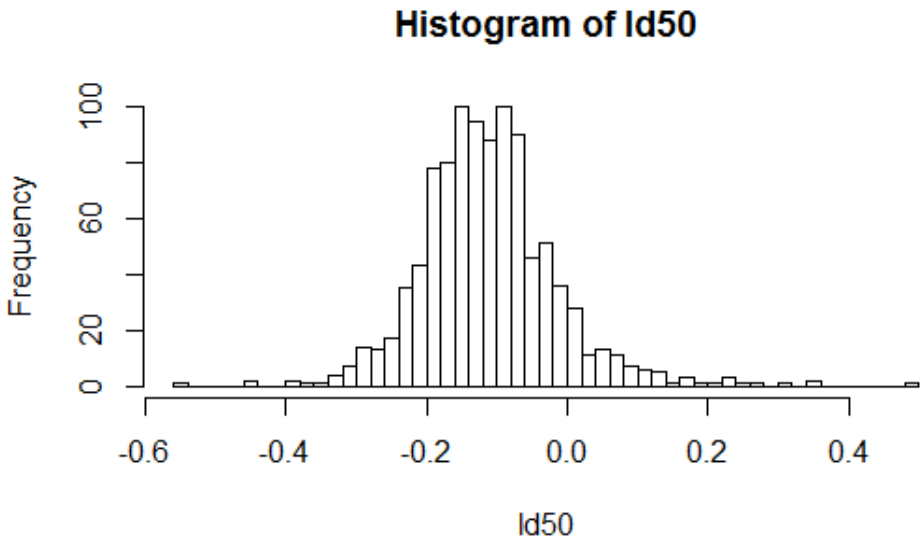
$for(iin(1:na))\{$
$z[i,] = posteriori(alpha[i], beta) * 1000$

$contour(alpha, beta, z, nlevels = 25)$
$ll = 200$
$alpha = seq(-4, 8, length = ll)$
$beta = seq(-10, 40, length = ll)$
$PP = matrix(NA, ll, ll)$
$for(iin1:ll)\{$
$for(jin1:ll)\{$
$PP[i,j] = posteriori(alpha[i], beta[j])$
$\}$
$\}$
$MM = max(PP)$
$ccc = sum(PP)$
$PP = PP/ccc$
$II = PP$
$NN = 1000$
$alpha.marginal < -apply(II, 1, sum)$
$beta.marginal < -apply(II, 2, sum)$
$ggg = numeric(NN)$
$ddd = numeric(NN)$
$ttt < -matrix(NA, length(x), NN)$
$for(lin1:NN)\{$
$uuu < -runif(1, 0, 1)$
$ggg[l] < -max(alpha[cumsum(alpha.marginal) < uuu])$
$junk < -(1:length(alpha))[alpha == ggg[l]]$
$conditional < -II[junk,]/sum(II[junk,])$
$uuu < -runif(1, 0, 1)$
$ddd[l] < -max(beta[cumsum(conditional) < uuu])$
$\}$
$d.alpha = alpha[2] - alpha[1]$
$d.beta = beta[2] - beta[1]$
$ggg = ggg + runif(length(ggg), 0, 1) * d.alpha - d.alpha/2$
$ddd = ddd + runif(length(ddd), 0, 1) * d.beta - d.beta/2$
$par(mfrow = c(1, 1))$
$plot(x = NULL,$
$y = NULL,$

$xlim = range(-5:10),$
$ylim = range(-5:30), xlab = "alpha", ylab = "beta"$
$)$
$points(x = ggg, y = ddd)$
$ld50 = -ggg/ddd$
$hist(ld50, breaks = 40, probability = FALSE)$




30
10
0
-10
-5
0
5
10
4e−04
beta
25
15
5
-5
-5
0
5
10
alpha


Histogram of ld50
Frequency
0
20
60
100
-0.6
-0.4
-0.2
0.0
0.2
0.4
ld50

0.3. Ejercicio 11.3

3. Gibbs sampling: Table 11.4 contains quality control measurements from 6 machines in a factory. Quality control measurements are expensive and time-consuming, so only 5 measurements were done for each machine. In addition to the existing machines, we are interested in the quality of another machine (the seventh machine). Implement a

---

Machine	Measurements
1	83, 92 ,92, 46, 67
2	117, 109, 114, 104, 87
3	101, 93, 92, 86, 67
4	105, 119, 116, 102, 116
5	79, 97, 103, 79, 92
6	57, 92, 104, 77, 100

Table 11.4: *Quality control measurements from 6 machines in a factory.*

separate, a pooled and hierarchical Gaussian model with common variance described in Section 11.6. Run the simulations long enough for approximate convergence. Using each of three models—separate, pooled, and hierarchical—report: (i) the posterior distribution of the mean of the quality measurements of the sixth machine, (ii) the predictive distribution for another quality measurement of the sixth machine, and (iii) the posterior distribution of the mean of the quality measurements of the seventh machine.

Solución

Podemos usar un modelo por separado para cada máquina o sea en este caso las mediciones para la máquina j tienen media θ_j

Para el modelo combinado se asume que todas las mediciones(de las 6 máquinas) tienen una media común.

Para el modelo gaussiano análogo al ejemplo 11.6, el algoritmo de Gibbs en R es :

$theta_update = function(){$

$theta_hat < -(mu/tau2+promedios*length.j/sigma2)/(1/tau2+length.j/sigma2)$

$V_theta < -1/(1/tau2 + length.j/sigma2)$

$result = numeric(J)$

$for(iin1 : J){$

$result[i] = rnorm(1, theta_hat[i], sqrt(V_theta[i]))$
$\}$
$return(result)$
$\}$
$mu_update = function()\{$
$mu.hat = mean(theta)$
$return(rnorm(1, mu.hat, sqrt(tau2/J)))$
$\}$
$sigma2_update = function()\{$
$theta.aux = c(rep(theta, length.j))$
$sigma2.hat = sum((datos\$mediciones - theta.aux)^2)/N$
$sigma2 = (1/rchisq(1, N)) * N * sigma2.hat$
$return(sigma2)$
$\}$
$tau2_update = function()\{$
$tau2.hat = sum((theta - mu)^2)/(J - 1)$
$tau2 = (1/rchisq(1, J - 1)) * (J - 1) * tau2.hat$
$return(tau2)$
$\}$
$log.posteriori.conjunta.menos.theta_update = function()\{$
$theta.aux = c(rep(theta, length.j))$
$sum = 0{,}5 * log(tau2) - 10$
$for(iin1 : N)\{$
$sum = sum + log(dnorm(datos\$mediciones[i], theta.aux[i], sqrt(sigma2)))$
$\}$
$return(sum)$
$\}$
$log.posteriori.conjunta.con.theta_update = function()\{$
$theta.aux = c(rep(theta, length.j))$
$sum = 0{,}5 * log(tau2)$
$for(iin1 : N)\{$
$sum = sum + log(dnorm(datos\$mediciones[i], theta.aux[i], sqrt(sigma2)))$
$\}$
$for(jin1 : J)\{$
$sum = sum + log(dnorm(theta[j], mu, sqrt(tau2)))$
$\}$
$return(sum)$
$\}$

$maquinas = as.factor(c(rep("1", 5), rep("2", 5), rep("3", 5), rep("4", 5), rep("5", 5), rep("6", 5)))$
$mediciones = c(83, 92, 92, 46, 67, 117, 109, 114, 104, 87, 101, 93, 92, 86,$
$67, 105, 119, 116, 102, 116, 79, 97, 103, 79, 92, 57, 92, 104,$
$77, 100)$
$datos = data.frame(maquinas = maquinas, mediciones = mediciones)$
$promedios = tapply(datos\$mediciones, datos\$maquinas, mean)$
$length.j = tapply(datos\$mediciones, datos\$maquinas, length)$
$J = length(levels(datos\$maquinas))$
$n.par = 9$
$N = length(datos[, 1])$
$chains < -10$
$iter < -100$
$sims < -array(NA, c(iter, chains, n.par + 2))$
$dimnames(sims) < -list(NULL, NULL,$
$c("theta1",$
$"theta2", "theta3", "theta4", "theta5", "theta6", "mu", "sigma2", "tau2", "logpost1", "logpost2"))$
$for(iin1 : chains)\{$
$theta = tapply(datos\$mediciones, datos\$maquinas, sample, size = 1)$
$mu = mean(theta)$
$sigma2 = sigma2_update()$
$tau2 = tau2_update()$
$log1 = log.posteriori.conjunta.menos.theta_update()$
$log2 = log.posteriori.conjunta.con.theta_update()$
$sims[1, i,] = c(theta, mu, sigma2, tau2, log1, log2)$

$for(min1 : chains)\{$
$for(tin1 : (iter - 1))\{$
$theta < -sims[t, m, 1 : J]$
$mu < -sims[t, m, J + 1]$
$sigma2 < -sims[t, m, J + 2]$
$tau2 < -sims[t, m, J + 3]$
$theta < -theta_update()$
$mu < -mu_update()$
$sigma2 < -sigma2_update()$
$tau2 < -tau2_update()$
$log1 = log.posteriori.conjunta.menos.theta_update()$
$log2 = log.posteriori.conjunta.con.theta_update()$
$sims[t + 1, m,] < -c(theta, mu, sigma2, tau2, log1, log2)$

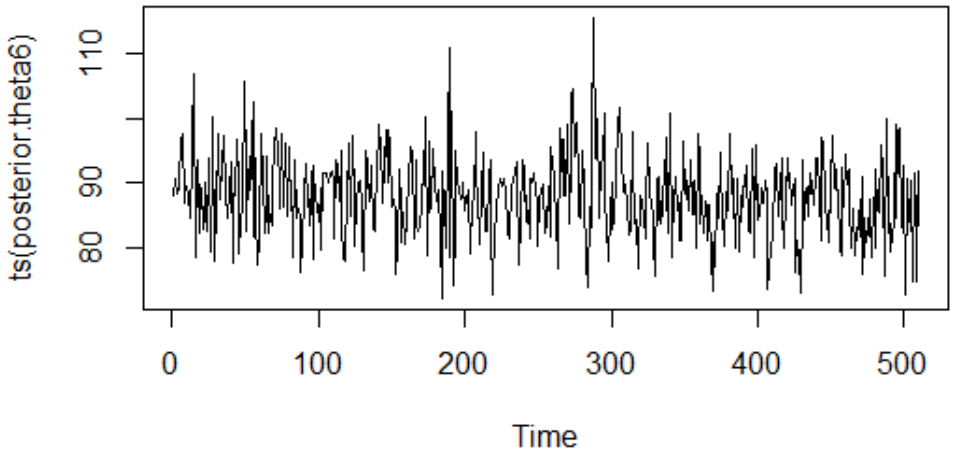
$$
\}
\}
posterior.theta1 = c(sims[floor(iter/2) : iter, 1 : chains, 1])
mean(posterior.theta1)
hist(posterior.theta1, probability = TRUE)
densidad = density(posterior.theta1)
lines(densidad, col = "red")
quantile(posterior.theta1, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
posterior.theta2 = c(sims[floor(iter/2) : iter, 1 : chains, 2])
mean(posterior.theta2)
quantile(posterior.theta2, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
posterior.theta3 = c(sims[floor(iter/2) : iter, 1 : chains, 3])
mean(posterior.theta3)
quantile(posterior.theta3, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
posterior.theta4 = c(sims[floor(iter/2) : iter, 1 : chains, 4])
mean(posterior.theta4)
quantile(posterior.theta4, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
posterior.theta5 = c(sims[floor(iter/2) : iter, 1 : chains, 5])
mean(posterior.theta5)
quantile(posterior.theta5, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
posterior.theta6 = c(sims[floor(iter/2) : iter, 1 : chains, 6])
mean(posterior.theta6)
quantile(posterior.theta6, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
posterior.mu = c(sims[floor(iter/2) : iter, 1 : chains, 7])
mean(posterior.mu)
quantile(posterior.mu, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
posterior.sigma = sqrt(c(sims[floor(iter/2) : iter, 1 : chains, 8]))
mean(posterior.sigma)
quantile(posterior.sigma, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
posterior.tau = sqrt(c(sims[floor(iter/2) : iter, 1 : chains, 9]))
mean(posterior.tau)
quantile(posterior.tau, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
log.posterior.sin.theta = (c(sims[floor(iter/2) : iter, 1 : chains, 10]))
mean(log.posterior.sin.theta)
quantile(log.posterior.sin.theta, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
log.posterior.con.theta = (c(sims[floor(iter/2) : iter, 1 : chains, 11]))
mean(log.posterior.con.theta)
quantile(log.posterior.con.theta, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))
$$

El comando monitor de la libreria rstan permite monitorear la convergencia

Inference for the input samples (10 chains: each with iter=100; warmup=50):

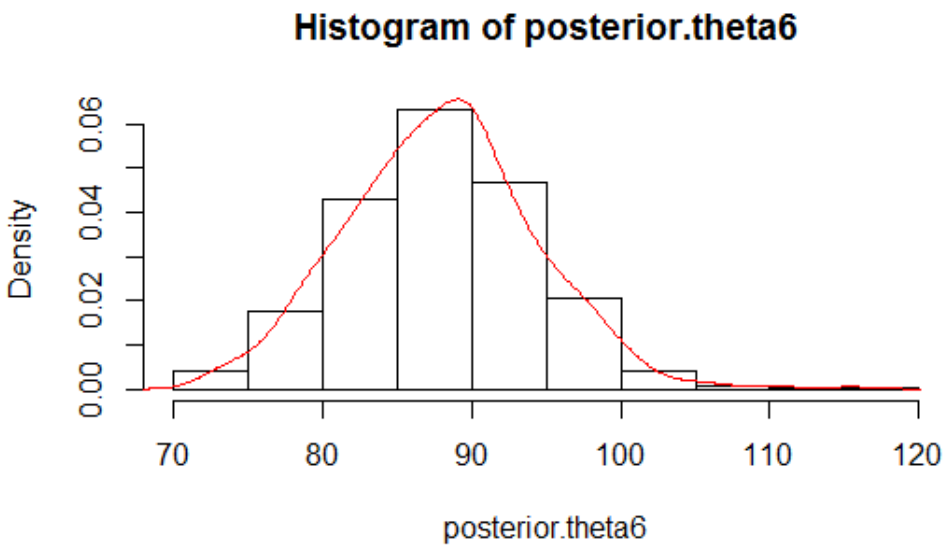
	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
theta1	80.0	0.5	7.2	65.5	75.2	80.0	85.3	93.0	236	1.0
theta2	103.2	0.4	6.6	91.2	98.5	103.1	107.8	116.3	273	1.0
theta3	88.9	0.3	5.9	76.5	84.9	88.8	93.1	100.7	387	1.0
theta4	107.6	0.5	6.9	94.0	102.8	107.9	112.2	121.7	220	1.1
theta5	91.1	0.3	6.1	78.8	87.7	90.9	95.0	102.1	427	1.0
theta6	88.0	0.3	6.4	76.0	83.8	88.0	91.7	100.3	384	1.0
mu	93.3	0.4	8.6	76.0	88.9	93.5	97.7	110.4	423	1.0
sigma2	231.6	4.3	73.9	127.3	179.2	217.2	272.3	411.9	297	1.0
tau2	396.7	42.5	587.0	13.0	111.6	221.4	473.1	1832.2	191	1.0
logpost1	-130.7	0.4	2.6	-136.8	-132.2	-130.2	-128.8	-127.2	42	1.1
logpost2	-144.7	0.2	2.6	-151.2	-145.9	-144.4	-143.0	-141.0	173	1.1

For each parameter, n_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat=1).

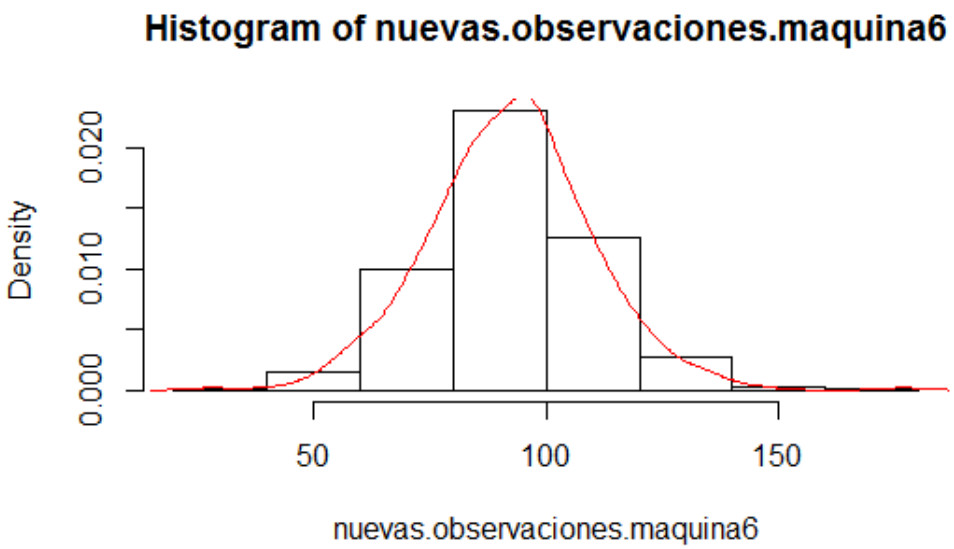


i)Con esta tabla la distribución a posteriori de la media de las mediciones de

calidad de la máquina 6: θ_6, se puede ver que el intervalo al 95 % para este valor es [76,0, 100,3]. La gráfica es



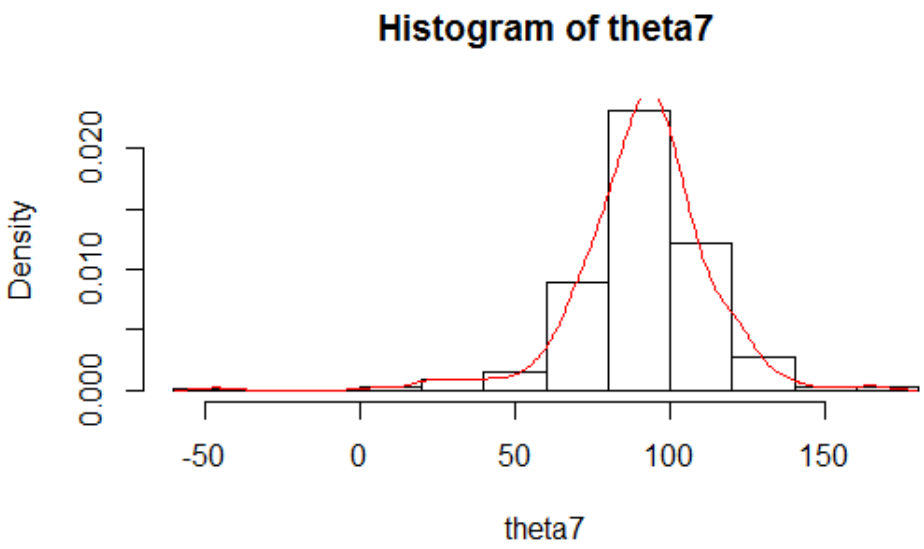
ii) La distribucion predictiva para una nueva medición(la sexta) de calidad de la máquina 6 $y_{6,6}^{\sim}$ es
$p(y_{6,6}^{\sim}|y) = \int p(y_{6,6}^{\sim}|\theta)p(\theta|y)d\theta = \int\int p(y_{6,6}^{\sim}|\theta_6, \sigma^2)p(\theta_6, \sigma^2|y)d\theta_6 d\sigma^2$. En R:
$nuevas.observaciones.maquina6 = numeric(length(posterior.theta6))$
$for(iin1 : length(posterior.theta6))\{$
$theta_6 = posterior.theta6[i]$
$sigma2 = posterior.sigma2[i]$
$nuevas.observaciones.maquina6[i] = rnorm(1, theta_6, sqrt(sigma2))$
$\}$
$hist(nuevas.observaciones.maquina6, probability = TRUE)$
$densidad = density(nuevas.observaciones.maquina6)lines(densidad, col =$
$"red")$
$quantile(nuevas.observaciones.maquina6, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))$
y la gráfica de la distribución predictiva es



y los cuantiles:
2,5 % 25 % 50 % 75 % 97,5 %
62.73723 82.78670 93.33072 103.03003 126.50737

iii) La distribución a posteriori de la media de las mediciones de calidad para una máquina 7: $\tilde{\theta_7}$ es
$p(\tilde{\theta_7}|y) = \int p(\tilde{\theta_7}|\theta)p(\theta|y)d\theta = \int\int p(\theta_7|\mu,\tau^2)p(\mu,\tau^2|y)d\mu d\tau^2$
En R:
$theta7 = numeric(length(posterior.mu))$
$for(iin1 : length(posterior.theta6))\{$
$mu = posterior.mu[i]$
$tau2 = posterior.tau2[i]$
$theta7[i] = rnorm(1, mu, sqrt(tau2))$
$\}$
$hist(theta7, probability = TRUE)$
$densidad = density(theta7)$
$lines(densidad, col = "red")$
$quantile(theta7, c(0{,}025, 0{,}25, 0{,}5, 0{,}75, 0{,}975))$

y se obtiene el gráfico



Y los cuantiles
2,5 % 25 % 50 % 75 % 97,5 %
42.05024 80.64965 92.57932 103.09243 128.37916

0.4. Ejercicio 11.4

> 4. Gibbs sampling: Extend the model in Exercise 11.3 by adding a hierarchical model for the variances of the machine quality measurements. Use an Inv-χ^2 prior distribution for variances with unknown scale σ_0^2 and fixed degrees of freedom. (The data do not contain enough information for determining the degrees of freedom, so inference for that hyperparameter would depend very strongly on its prior distribution in any case). The conditional distribution of σ_0^2 is not of simple form, but you can sample from its distribution, for example, using grid sampling.

Solución
Tenemos el modelo jerárquico:
$y_{ij} \sim N(\theta_j, \sigma_j^2)$ para $i = 1, \ldots, n_j$ y $j = 1, \ldots, 6$,
$\theta_j \sim N(\mu, \tau^2)$ y
$\sigma_j^2 \sim inv - \chi^2(\nu, \sigma_0^2)$
con hiper parámetros $(\mu, \tau^2, \nu, \sigma_0^2)$ y ν es fijo y σ_0^2desconocido Como no tenemos suficientes datos para estimar los grados de libertad podemos hacer el

método de la grilla .
La distribución a posteriori conjunta es(la a apriori de τ se mantiene uniforme)
$p(\theta, \mu, \sigma^2, log\tau|y) \propto$
$\tau \prod_{j=1}^{J} (\frac{\sigma_0^2}{\sigma_j^2})^{\nu/2+1} exp(-\frac{\nu\sigma_0^2}{2\sigma_j^2}) \prod_{j=1}^{J} N(\theta_j|\mu, \tau^2) \prod_{j=1}^{J} \prod_{i=1}^{n_j} N(y_{ij}|\theta_j, \sigma_j^2)$
$\propto \tau(\sigma_0^2)^{J(\nu/2+1)} \prod_{j=1}^{J} (\frac{1}{\sigma_j^2})^{\nu/2+1} exp(-\frac{\nu\sigma_0^2}{2\sigma_j^2}) \prod_{j=1}^{J} N(\theta_j|\mu, \tau^2) \prod_{j=1}^{J} \prod_{i=1}^{n_j} N(y_{ij}|\theta_j, \sigma_j^2)$
La distribución condicional de los θ_j solo cambia en que el σ común ahora cambia a σ_j, la condicional de μ es la misma, la condicional de τ^2 es la misma y la condicional a posteriori para σ_j^2 es
$\sigma_j^2|y, \nu, \sigma_0^2, \theta_j, \mu, \tau^2 \sim inv-\chi^2(\nu+n_j, \frac{\nu\sigma_0^2+n_j v_j}{\nu+n_j})$ donde $v_j = \frac{1}{n_j} \sum_{i=1}^{n_j} (y_{ij} - \theta_j)^2$.
Los valores de ν se escogieron aleatoriamente para cada cadena(se simularon 10 cadenas) en $[5, 10]$.En R
$theta_update = function()\{$
$theta_hat < -(mu/tau2 +$
$promedios * length.j/sigma2)/(1/tau2 + length.j/sigma2)$
$V_theta < -1/(1/tau2 + length.j/sigma2)$
$result = numeric(J)$
$for(iin1 : J)\{$
$result[i] = rnorm(1, theta_hat[i], sqrt(V_theta[i]))$
$\}$
$return(result)$
$\}$
$mu_update = function()\{$
$mu.hat = mean(theta)$
$return(rnorm(1, mu.hat, sqrt(tau2/J)))$
$\}$
$sigma2_update = function()\{$
$theta.aux = c(rep(theta, length.j))$
$temp = (datos\$mediciones - theta.aux)^2$
$temp2 = tapply(temp, rep(1 : J, length.j), sum)$
$v = temp2/length.j$
$sigma2 = numeric(J)$
$for(iin1 : J)\{$
$sigma2[i] = (1/rchisq(1, nu+length.j[i])) * (nu * sigma2_0 + length.j[i] * v[i])$
$\}$
$return(sigma2)$

$$
\}
$$
$$
tau2_upd ate = function()\{
$$
$$
tau2.hat = sum((theta - mu)^2)/(J - 1)
$$
$$
tau2 = (1/rchisq(1, J - 1)) * (J - 1) * tau2.hat
$$
$$
return(tau2)
$$
$$
\}
$$
$$
sigma2_{0u}pdate = function()\{
$$
$$
ll = 100
$$
$$
sigma2_0.array = seq(4, 100, length = ll)
$$
$$
PP = numeric(ll)
$$
$$
for(iin1 : ll)\{
$$
$$
PP[i] = log.posteriori.conjunta(nu, sigma2_0.array[i])
$$
$$
\}
$$
$$
MM = max(PP)
$$
$$
PP = exp(PP - MM)
$$
$$
ccc = sum(PP)
$$
$$
PP = PP/ccc
$$
$$
II = PP
$$
$$
NN = 20
$$
$$
muestras.sigma2_0 = sample(sigma2_0.array, size = 20, prob = II)
$$
$$
return(mean(muestras.sigma2_0))
$$
$$
\}
$$
$$
log.posteriori.conjunta = function(nu, sigma2_0.par)\{
$$
$$
theta.aux = c(rep(theta, length.j))
$$
$$
sigma2.aux = c(rep(sigma2, length.j))
$$
$$
sum = 0{,}5 * log(tau2) + J * (nu/2 + 1) * (log(sigma2_0.par))
$$
$$
sum = sum + sum(log(dnorm(datos\$mediciones, theta.aux, sqrt(sigma2.aux))))
$$
$$
sum = sum - sum((nu/2
$$
$$
+ 1) * log(sigma2)) - sum((nu * sigma2_0.par/(2 * sigma2)))
$$
$$
+ sum(log(dnorm(theta, rep(mu, J), sqrt(rep(tau2, J)))))
$$
$$
return(sum)
$$

$$
maquinas = as.factor(c(rep("1", 5), rep("2", 5), rep("3", 5), rep("4", 5), rep("5", 5), rep("6", 5)))
$$
$$
mediciones = c(83, 92, 92, 46, 67, 117, 109, 114, 104, 87, 101, 93, 92, 86,
$$
$$
67, 105, 119, 116, 102, 116, 79, 97, 103, 79, 92, 57, 92, 104,
$$
$$
77, 100)
$$
$$
datos = data.frame(maquinas = maquinas, mediciones = mediciones)
$$
$$
promedios = tapply(datos\$mediciones, datos\$maquinas, mean)
$$

$length.j = tapply(datos\$mediciones, datos\$maquinas, length)$
$J = length(levels(datos\$maquinas))$
$n.par = 14$
$N = length(datos[, 1])$
$chains < -10$
$iter < -1000$
$sims < -array(NA, c(iter, chains, n.par + 2))$
$dimnames(sims) < -list(NULL, NULL,$
$c("theta1",$
$"theta2","theta3","theta4","theta5","theta6","mu","sigma2_1","sigma2_2","sigma2_3",$
$"sigma2_4","sigma2_5","sigma2_6","tau2","sigma2_0","logpost2"))$
$nu.array = sample(5 : 10, size = chains, replace = TRUE)$
$for(iin1 : chains)\{$
$theta = tapply(datos\$mediciones, datos\$maquinas, sample, size = 1)$
$nu = nu.array$
$mu = mean(theta)$
$tau2 = tau2_update()$
$sigma2_0 = runif(1, 4, 100)$
$sigma2 = sigma2_update()$
$sigma2_0 = sigma2_{0u}pdate()$
$log3 = log.posteriori.conjunta(nu, sigma2_0)$
$sims[1, i,] = c(theta, mu, sigma2, tau2, sigma2_0, log3)$
$\}$
$for(min1 : chains)\{$
$nu = nu.array[m]$
$for(tin1 : (iter - 1))\{$
$theta < -sims[t, m, 1 : J]$
$mu < -sims[t, m, J + 1]$
$sigma2 < -sims[t, m, (J + 2) : (2 * J + 1)]$
$tau2 < -sims[t, m, 2 * J + 2]$
$sigma2_0 < -sims[t, m, 2 * J + 3]$
$theta < -theta_update()$
$mu < -mu_update()$
$tau2 < -tau2_update()$
$sigma2 < -sigma2_update()$
$sigma2_0 = sigma2_{0u}pdate()$
$log1 = log.posteriori.conjunta.menos.theta_update()$
$log2 = log.posteriori.conjunta.con.theta_update()$

$log3 = log.posteriori.conjunta(nu, sigma2_0)$
$sims[t+1, m,] < -c(theta, mu, sigma2, tau2, sigma2_0, log3)\}$
$\}$
Con lo que se obtiene

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
theta1	79.5	0.1	7.0	66.3	74.8	79.3	83.9	94.2	2979	1.0
theta2	104.7	0.1	5.1	94.3	101.6	104.8	108.0	114.3	3696	1.0
theta3	88.5	0.1	5.0	78.7	85.3	88.5	91.6	98.8	4656	1.0
theta4	109.9	0.1	4.5	100.6	107.3	110.0	112.8	118.1	3135	1.0
theta5	90.4	0.1	4.7	81.3	87.4	90.4	93.4	99.9	4339	1.0
theta6	87.6	0.1	6.3	75.3	83.4	87.6	91.6	100.4	4147	1.0
mu	93.5	0.1	9.0	75.4	89.1	93.5	98.1	110.2	4464	1.0
sigma2_1	261.6	13.8	161.9	100.8	164.6	221.1	310.0	669.7	139	1.0
sigma2_2	136.6	1.3	81.4	53.0	86.7	117.4	161.2	345.2	4078	1.0
sigma2_3	146.4	1.3	85.5	56.7	94.0	125.5	174.1	347.9	4070	1.0
sigma2_4	97.3	1.0	58.4	37.1	60.9	82.3	115.2	245.3	3292	1.0
sigma2_5	124.7	1.1	71.6	48.8	79.5	107.5	147.5	305.6	4305	1.0
sigma2_6	243.8	2.8	142.5	94.9	156.3	208.2	287.9	600.0	2564	1.0
tau2	423.3	17.9	899.5	49.1	133.3	235.0	435.3	1886.8	2533	1.0
sigma2_0	88.8	0.0	1.4	85.4	88.2	89.1	89.8	90.6	2030	1.0
logpost2	-168.4	2.1	5.8	-180.8	-172.6	-167.5	-163.8	-159.4	8	1.7

For each parameter, n_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat=1).

0.5. Ejercicio 11.5

5. Monitoring convergence:
   (a) Prove that $\widehat{\text{var}}^+(\psi|y)$ as defined in (11.3) is an unbiased estimate of the marginal posterior variance of ϕ, if the starting distribution for the Markov chain simulation algorithm is the same as the target distribution, and if the m parallel sequences are computed independently. (Hint: show that $\widehat{\text{var}}^+(\psi|y)$ can be expressed as the average of the halved squared differences between simulations ϕ from different sequences, and that each of these has expectation equal to the posterior variance.)
   (b) Determine the conditions under which $\widehat{\text{var}}^+(\psi|y)$ approaches the marginal posterior variance of ϕ in the limit as the lengths n of the simulated chains approach ∞.

Solución

- (a)

0.6. Ejercicio 11.6

6. Effective sample size:
   (a) Derive the asymptotic formula (11.5) for the variance of the average of correlated simulations.
   (b) Implement a Markov chain simulation for some example and plot $\widehat{n}_{\text{eff}}$ from (11.8) over time. Is $\widehat{n}_{\text{eff}}$ stable? Does it gradually increase as a function of number of iterations, as one would hope?

Solución

- (a)

0.7. Ejercicio 11.7

7. Analysis of survey data: Section 8.3 presents an analysis of a stratified sample survey using a hierarchical model on the stratum probabilities.
   (a) Perform the computations for the simple nonhierarchical model described in the example.
   (b) Using the Metropolis algorithm, perform the computations for the hierarchical model, using the results from part (a) as a starting distribution. Check by comparing your simulations to the results in Figure 8.1b.

Solución

- (a)

0.8. Bibliografía

[1] http://www.stat.columbia.edu/ gelman/book/solutions2.pdf
[2] http://www.stat.columbia.edu/ gelman/book/solutions3.pdf
[3] http://www.stat.columbia.edu/ gelman/book/slides/class3b.pdf
[4] http://streylab.com/blog/2014/3/21/gelman-bioassay-without-mcmc
[5] http://www.stat.missouri.edu/ dsun/8640/Figure5.3a
[6] http://www.stat.missouri.edu/ dsun/8640/bioassay3.R
[7] https://classes.soe.ucsc.edu/ams207/Spring04/slide2.4.pdf

[8] http://www.stat.columbia.edu/ gelman/bugsR/software.pdf
[9] http://www.biostat.jhsph.edu/ fdominic/teaching/BM/tarone.S
[10] http://www.lce.hut.fi/teaching/S-114.2601/ex/exercises.shtml