

MATH 680 - Assignment #5

Kevin McGregor

December 21st, 2015

Question 1

(a)

First we find the posterior distribution of the parameters:

$$f(\beta, u, v_u | y) \propto \frac{1}{v_u^{n/2}} \exp \left\{ \frac{-1}{2v_u} \sum_{i=1}^n u_i^2 \right\} v_u^{-(\alpha_0-1)} e^{-\beta_0/v_u} \exp \left\{ \frac{-1}{2v_B} \sum_{j=1}^p \beta_j^2 \right\} \\ \prod_{i=1}^n \prod_{t=1}^{T_i} \left(\frac{\exp(x_{it}^\top \beta + u_i)}{1 + \exp(x_{it}^\top \beta + u_i)} \right)^{y_{it}} \left(\frac{1}{1 + \exp(x_{it}^\top \beta + u_i)} \right)^{1-y_{it}}.$$

The number of parameters we need to estimate is $p + n + 1$, i.e., p elements of β , n elements of u_i , and v_u . Our trial distribution on the k^{th} iteration will be:

$$N_{p+n+1} \left(\begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{p+n+1} \end{bmatrix}, \begin{bmatrix} s_1^2 & 0 & \dots & 0 \\ 0 & s_2^2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \dots & & s_{p+n+1}^2 \end{bmatrix} \right) \quad (1)$$

For convenience, we take the log of the posterior distribution:

$$\log f(\beta, u, v_u | y) = -\frac{n}{2} \log(v_u) - \frac{1}{2v_B} \sum_{j=1}^p \beta_j^2 - (\alpha_0 - 1) \log(v_u) - \frac{\beta_0}{v_u} - \frac{1}{2v_B} \sum_{j=1}^p \beta_j^2 \\ + \sum_{i=1}^n \sum_{t=1}^{T_i} \left\{ y_{it} \log \left(\frac{\exp(x_{it}^\top \beta + u_i)}{1 + \exp(x_{it}^\top \beta + u_i)} \right) + (1 - y_{it}) \log \left(\frac{1}{1 + \exp(x_{it}^\top \beta + u_i)} \right) \right\}.$$

If $v_u < 0$ then set $\log f = -\infty$. Next define:

$$\log \alpha \{(\beta_k, u_k, v_{uk}), (\beta_j, u_j, v_{uj})\} = \min \{0, \log f(\beta_j, u_j, v_{uj}) - \log f(\beta_k, u_k, v_{uk})\}.$$

So, to begin the algorithm, we choose initial values β_0 , u_0 , and v_{u0} , and set $k = 0$. The algorithm proceeds as follows:

1. Generate Z as specified in 1, and $U \sim U(0, 1)$.
2. If $\log U < \log \alpha \{(\beta_k, u_k, v_{uk}), Z\}$, set $(\beta_{k+1}, u_{k+1}, v_{u(k+1)}) = Z$. Otherwise set $(\beta_{k+1}, u_{k+1}, v_{u(k+1)}) = (\beta_k, u_k, v_{uk})$.
3. Set $k = k + 1$ and go to step (1).

(b)

The code for the R function for part (a) can be found in `bern_mcmc.R`

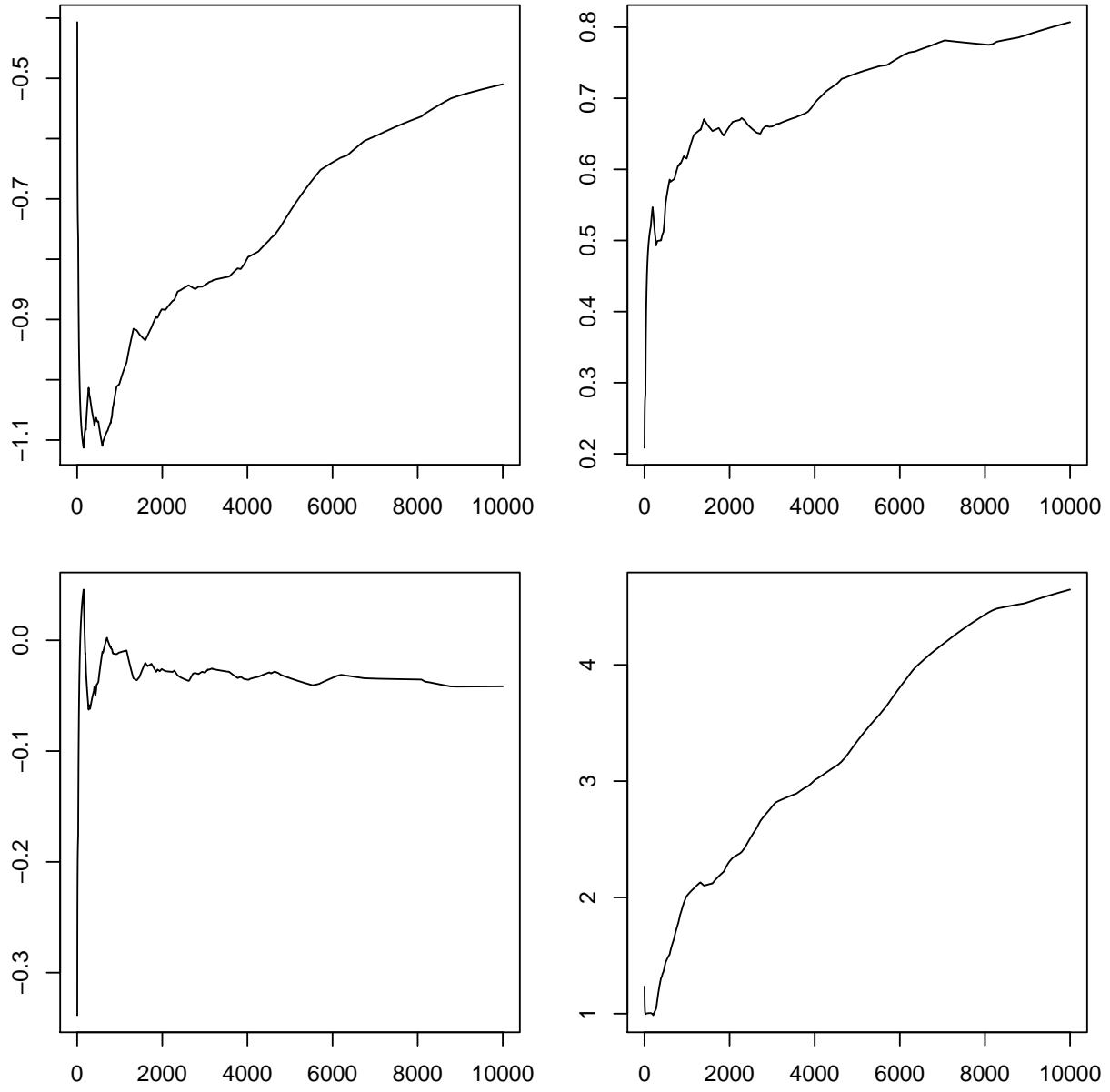


Figure 1: Trace plots for the elements of β and v_u (Part (c))

(c)

Code for this part can be found in the file *a5.q1.R*. The trace plots for this function can be found in Figure 1. As we can see, the trace plots do not look like convergence has been achieved yet; estimates could be improved. This is unsurprising due to the inefficiency of the algorithm. This algorithm would need to perform many more samples before proper convergence would be achieved. Estimated standard errors for $E(B|Y = y)$ are (0.29, 0.15, 0.07).

(d)

Code for this part can be found in the file *a5.q1.R*. The values I chose for this problem are: $n = 100$, $p = 2$, $T_i = 15$ for all i , $\beta = (1, 2, -2)^\top$, $v_u = 1$. Trace plots for this example can be found in Figure 2, and as before, convergence has not been reached and estimates could be improved.

Question 2

(a)

Code for this part can be found in the file *fusedRidge_crossval.R*.

(b)

$$f(\beta, v, \lambda|y) \propto \frac{1}{v^{n/2}} \exp \left\{ -\frac{1}{2v} (y - X\beta)^\top (y - X\beta) \right\} \frac{\lambda^{p/2}}{v^{p/2}} \exp \left\{ -\beta^\top \frac{\lambda}{2v} (wI_p + (1-w)H)\beta \right\} \\ v^{-a_v-1} e^{-b_v/v} \lambda^{a_L-1} e^{-b_L \lambda}.$$

Now,

$$f(\beta|v, \lambda, y) \propto \exp \left\{ -\frac{1}{2v} (y^\top y - 2\beta^\top X^\top y + \beta^\top X^\top X\beta + \beta^\top \lambda w I_p \beta + \beta^\top \lambda (1-w)H\beta) \right\} \\ \propto \exp \left\{ -\frac{1}{2v} \beta^\top (X^\top X + \lambda w I_p + \lambda(1-w)H) \beta + \frac{1}{v} \beta^\top (X^\top y) \right\}.$$

Therefore,

$$(B|V = v, L = \lambda, Y = y) \sim N_p \left((X^\top X + \lambda w I_p + \lambda(1-w)H)^{-1} (X^\top y), v (X^\top X + \lambda w I_p + \lambda(1-w)H)^{-1} \right).$$

Then, if we let $\mu_B = (X^\top X + \lambda w I_p + \lambda(1-w)H)^{-1} (X^\top y)$, we have:

$$f(\beta|v, \lambda, y) \propto_{\beta, v} \exp \left\{ -\frac{1}{2v} (\beta^\top (X^\top X + \lambda w I_p + \lambda(1-w)H) \beta - 2\beta^\top (X^\top y) + \mu_B^\top (X^\top X + \lambda w I_p + \lambda(1-w)H) \mu_B) \right\},$$

and so,

$$f(\beta, v|\lambda, y) \propto_{\beta, v} v^{-(n/2+p/2+a_v+1)} \exp \left\{ \frac{1}{2v} (y^\top y - 2\beta^\top X^\top y + \beta^\top X^\top X\beta + \lambda\beta^\top (wI_p + (1-w)H)\beta + 2b_v) \right\} \\ = f(\beta|v, \lambda, y) v^{-(n/2+a_v+1)} \exp \left\{ -\frac{1}{2v} [y^\top y - \mu_B^\top (X^\top X + \lambda w I_p + \lambda(1-w)H) \mu_B + 2b_v] \right\}.$$

Therefore,

$$(V|L = \lambda, Y = y) \sim \text{InvGam} \left(a_v + \frac{n}{2}, b_v + \frac{1}{2} [y^\top y - \mu_B^\top (X^\top X + \lambda w I_p + \lambda(1-w)H) \mu_B + 2b_v] \right).$$

Finally,

$$f(\lambda|\beta, v, y) \propto \lambda^{p/2+a_L-1} \exp \left\{ -\lambda \left(\frac{1}{2v} \beta^\top (wI_p + (1-w)H)\beta + b_L \right) \right\},$$

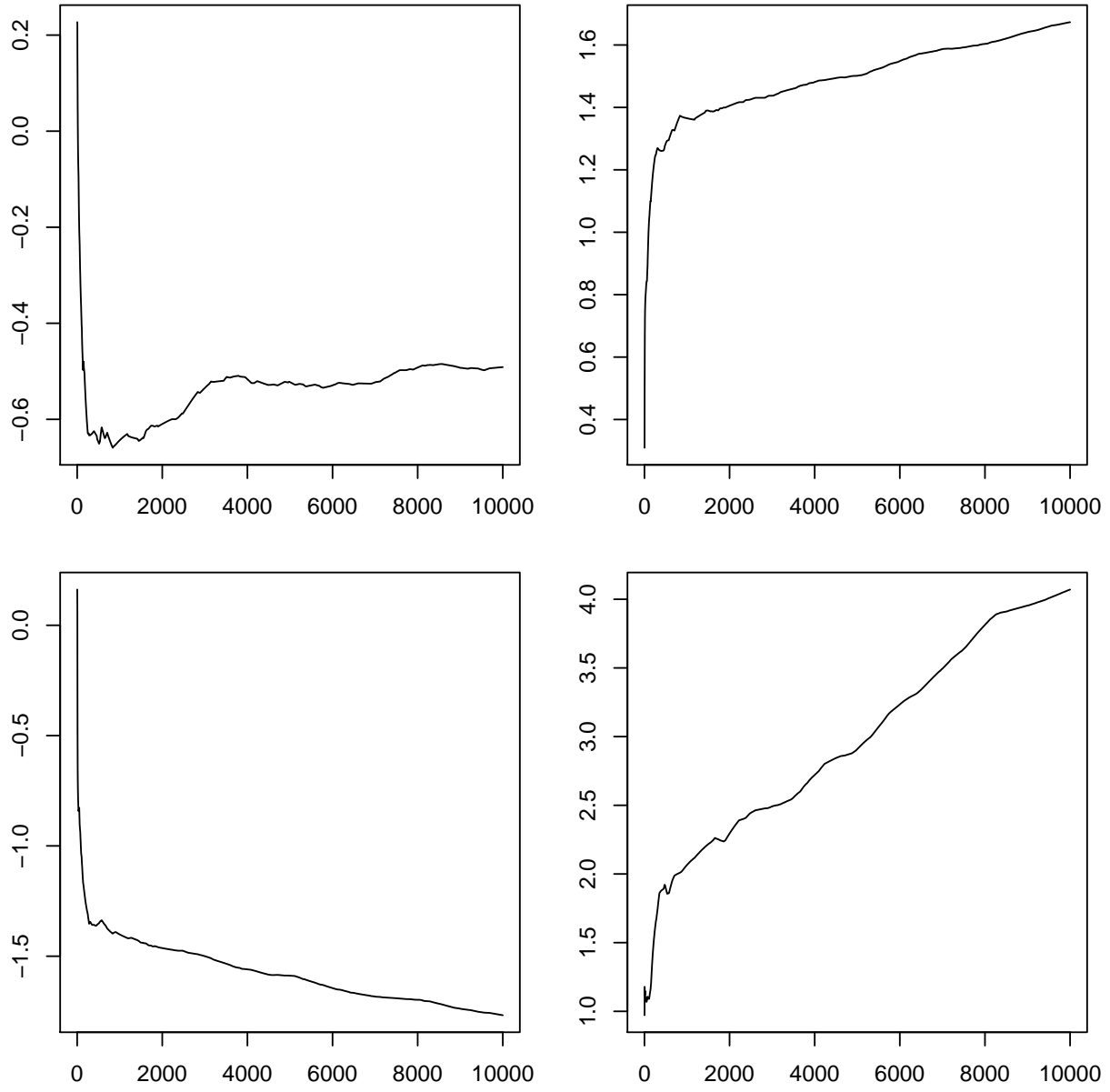


Figure 2: Trace plots for the elements of β and v_u (Part (d))

and so,

$$(L|B = \beta, V = v, Y = y) \sim \text{Gamma}\left(\frac{p}{2} + a_l, \frac{1}{2v}\beta^\top(wI_p + (1-w)H)\beta + b_L\right).$$

The algorithm begins by picking $(\beta_0, v_0, \lambda_0)$, and setting $k = 0$. The algorithm proceeds as follows:

1. Generate v_{k+1} from $(V|L = \lambda_k, Y = y)$ and β_{k+1} from $(B|V = v_{k+1}, L = \lambda_{k+1}, Y = y)$.
2. Generate λ_{k+1} from $(L|B = \beta_{k+1}, V = v_{k+1}, Y = y)$.
3. Set $k = k + 1$, and go to step (1).

Running the algorithm in R (*see file a5_q2bii.R*), we see estimates for beta and credible intervals in Table 1

Truth	Estimate	Lower	Upper
1.0000	0.8274	0.1305	1.5504
1.0000	0.9986	0.1586	1.8350
0.7500	0.9400	0.0409	1.8285
0.5000	-0.0318	-1.0351	0.9343
0.5000	0.9247	0.2412	1.5966

Table 1: Estimated values and credible intervals for β

We can also see the trace plots in Figure 3. Convergence appears to have been reached for all elements. Figure 4 shows the autocorrelation of the mean of the elements of beta over the different iterations. The autocorrelation is very small, even for smaller lags.

Question 3

(a)

We use Metropolis-Hastings sampling to sample from $\text{Beta}(\alpha, \beta)$ with a uniform trial density.

If $X \sim \text{Beta}(\alpha, \beta)$, then:

$$f_X(x) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}, \text{ for } x \in (0, 1).$$

Then, for some $b \in (0, 1)$, sample from the following trial distribution at every sample x_k :

$$Z \sim U(x_k - b, x_k + b)$$

Now, let

$$\alpha(x, y) = \min \left\{ 1, \left(\frac{y}{x} \right)^{\alpha-1} \left(\frac{1-y}{1-x} \right)^{\beta-1} \right\}, \text{ for } y \in (0, 1).$$

So, simply generate $U \sim U(0, 1)$, and accept the new sample (set $x_{k+1} = Z$) if $U < \alpha(x_k, Z)$.

(b)

Testing for this function (as well as the function itself) can be found in the file sampleBeta.R. Trace plots for two runs of this function (on different parameter values), can be seen in Figure 5. Convergence to the theoretical mean has been achieved in both cases.

We can also look at the autocorrelation plots for these two parameter scenarios. This can be seen in Figure 6. The autocorrelation declines rapidly, but seems to alternate between very small positive and negative correlations. Perhaps this has something to do with the boundedness of the beta distribution's support.

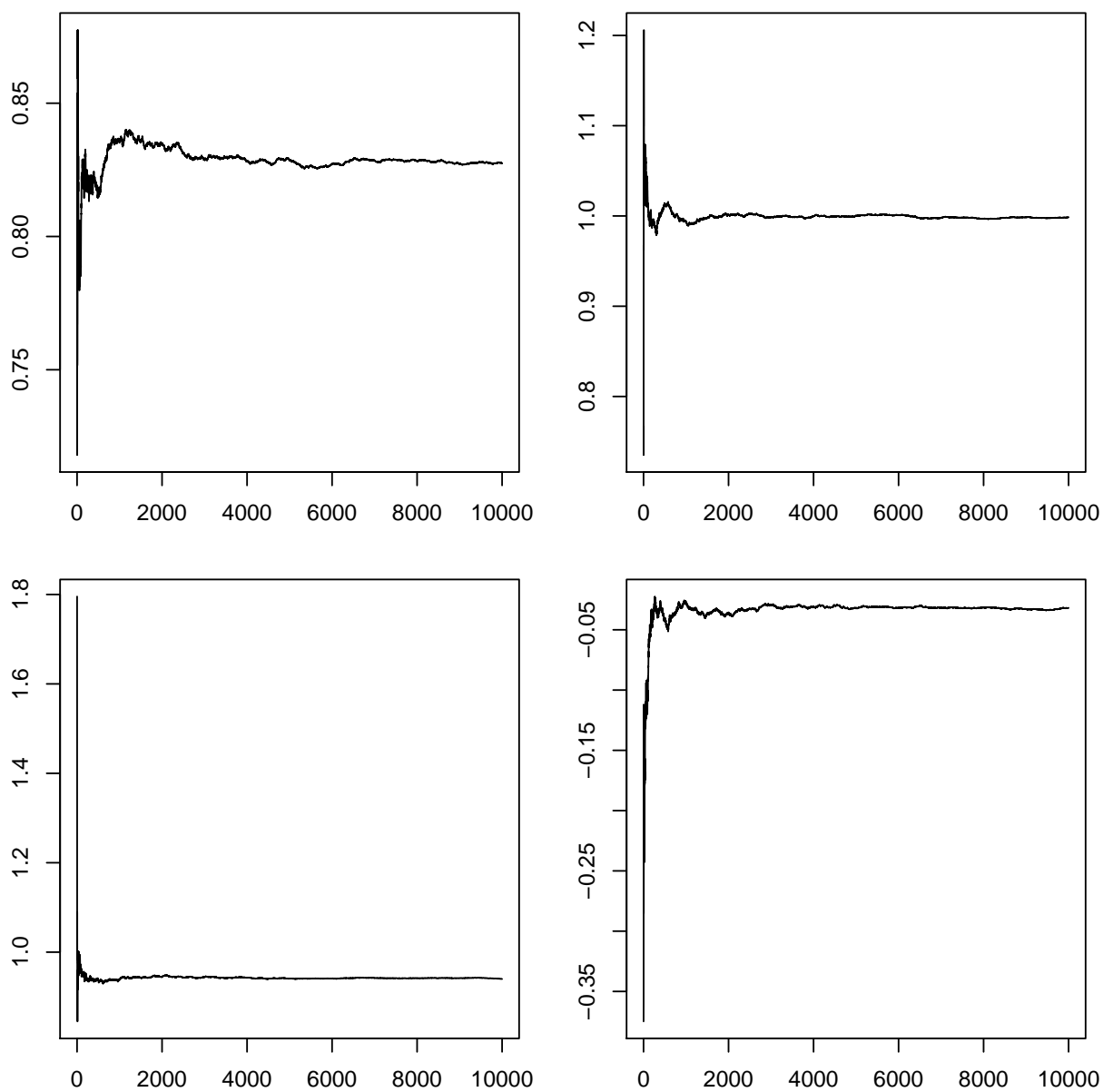


Figure 3: Trace plots for the first 4 elements of β

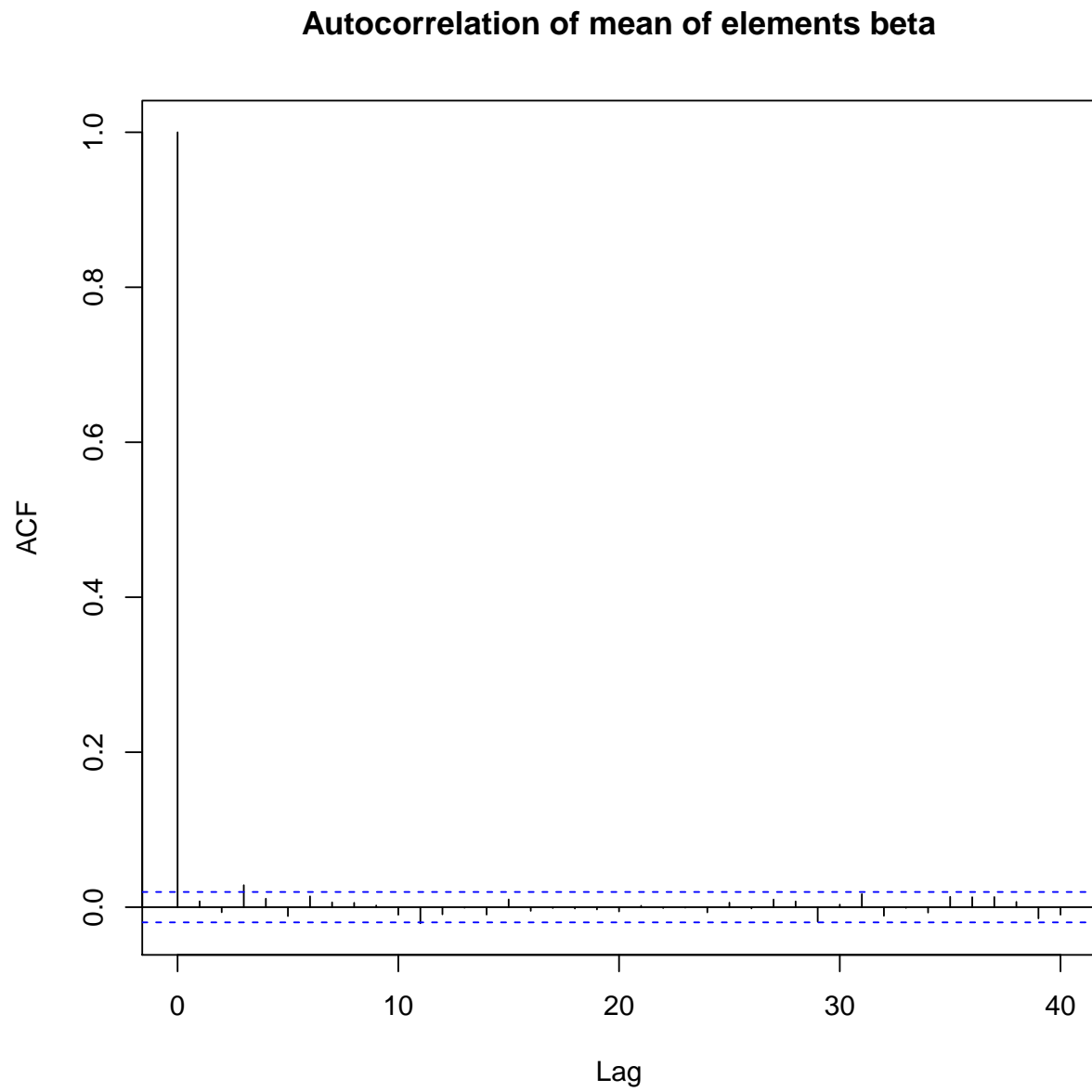


Figure 4: Plotting the autocorrelation for the mean of the estimated elements of beta

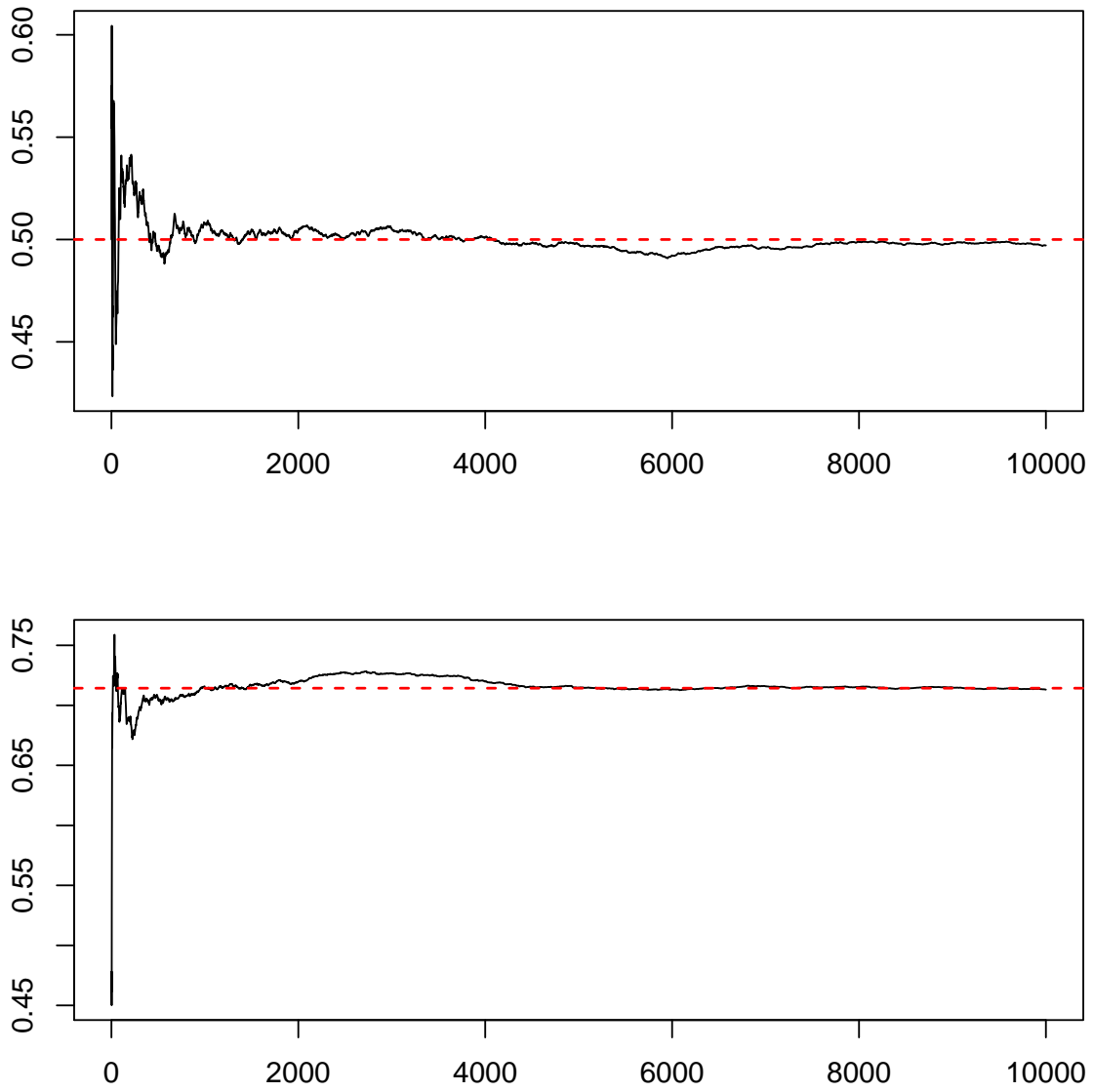
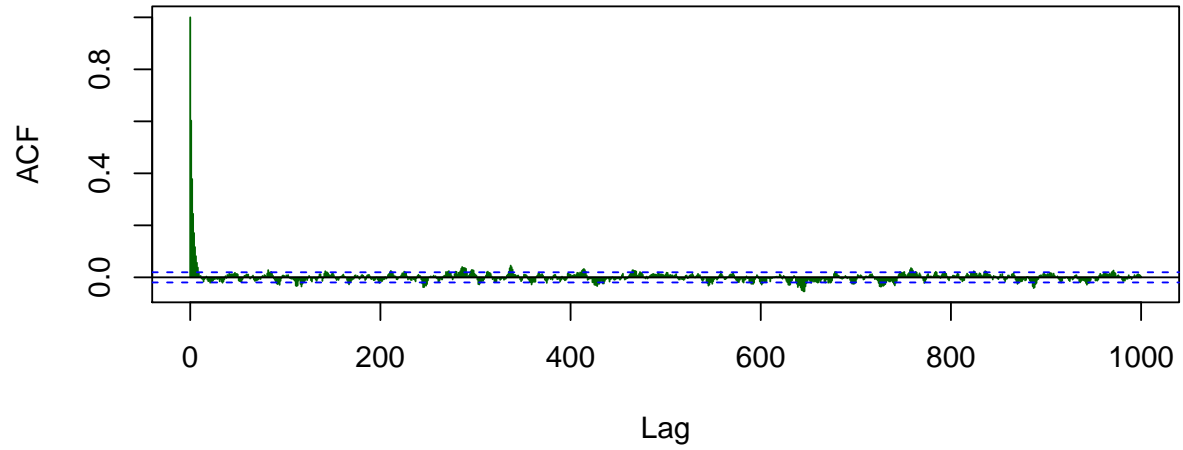


Figure 5: Trace plots for the means of two samples for the beta distribution (two different sets of parameters). The red line represents the theoretical mean

Autocorrelation: Beta MH sampling alpha=2, beta=2



Autocorrelation: Beta MH sampling alpha=5, beta=2

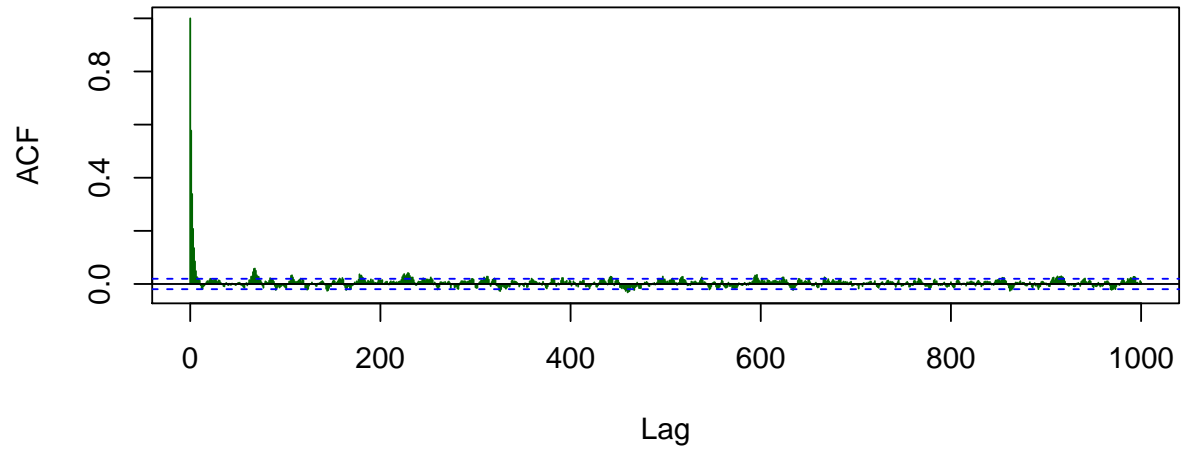


Figure 6: Autocorrelation functions for two parameter scenarios for sampling from the beta distribution.