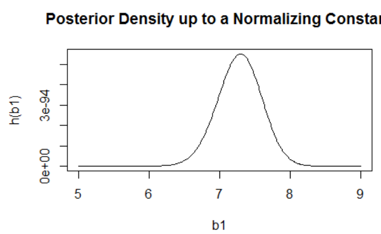
STAT 515 Takehome Final
Meng Chen

1. (a) The likelihood of $(\beta_1|Y, X, \sigma_i, \lambda) \propto \prod_{i=1}^{n} f(Y_i; \beta_0 + \beta_1 X_i, \sigma_i, \lambda)$ where f is the pdf for EMG.
The prior of β_1 is $N(0, 10)$
Hence the posterior $\pi(\beta_1|Y, X, \beta_0, \sigma_i, \lambda) \propto h(\beta_1) = \mathcal{L}(\beta_1|Y, X, \beta_0, \sigma_i, \lambda) \times f_{prior}(\beta_1)$

Figure 1.



For the Metropolis-Hastings algorithm, I selected a proposal density $q(\beta|\beta(t)) \sim N(\beta(t), 0.5^2)$, since the normal density centered at current value satisfies: 1) $q(x, y) = 0 \implies q(y, x) = 0$ (because of symmetry and fixed sd); 2) $q(x, y)$ is the transition kernal of an irreducaible Markov chain.

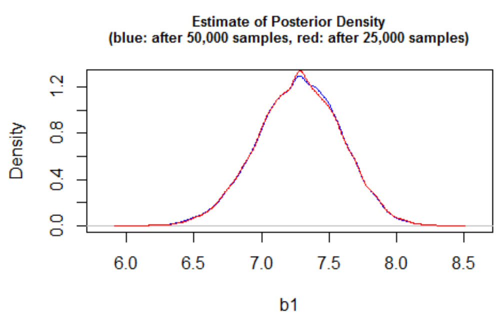
To sample from $\pi(\beta_1|Y, X)$ n times:
Step 0: Start with $\beta_1(0)$ randomly sampled from unif$(6, 8)$ in which range the posterior density is above zero by visual speculation. (Figure 1)
Repeat the following steps for t from 0 to (n-1) to get $\beta_1(1), ..., \beta_1(n)$:
Step 1: Generate a candidate b^* from $q(\beta_1|\beta_1(t))$
Step 2: Calculate acceptance probability $p = min(1, \frac{h(b^*)}{h(\beta_1(t))})$ ($q's$ cancell out due to symmetry)
Step 3: Generate a random u from unif$(0, 1)$
Step 4: If $u < p$, then $\beta_1(t + 1) = b^*$. Else $\beta_1(t + 1) = \beta_1(t)$

(b) I ran this sampler for a sample size of 50,000.
Estimate of the posterior expectation of β_1 = mean$(\beta_1(i))$ = 7.28. MCMC$_{SE}$=0.003

(c) The 95% credible interval for β_1 is (6.65, 7.86)

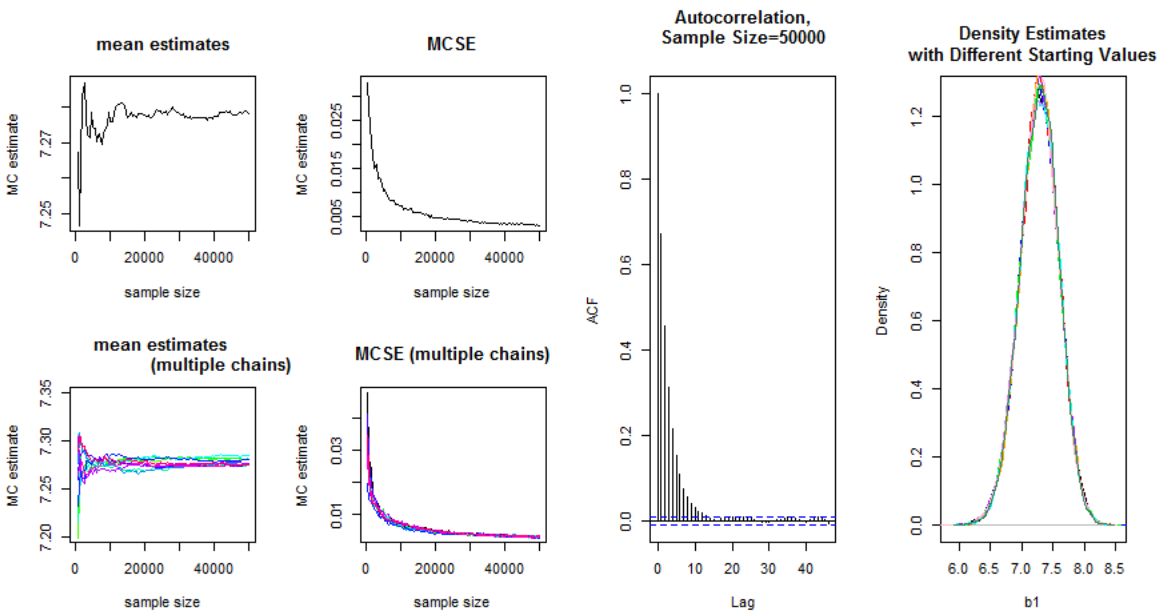
(d) Posterior Density Estimate in Figure 2.

Figure 2.



(e) First we look at the result from one chain (mean and MCSE reported in the above questions). The effective sample size is 9283, which means this chain is equivalent to about 9283 independent samples from π. By the rule of thumb, it is a sufficient sample size for estimate of expectation and confidence intervals. Also, as we can see in the plots for only one chain (the upper left two in Figure 3.), the fluctuation in the estimate of mean decreases as chain grows longer, and the Monte Carlo standard error also decreases and is very small when sample size approaches 50,000. The autocorrelations look reasonable as lag increases (Autocorrelation Plot in Figure 3). It seems like this chain mixed well. Then we will look at results from multiple chains with different starting values. I ran 10 chains with starting values from 6.2 to 8 with the increment of .2. As we can see in the plots (the lower left two in Figure 3), the mean estimates stabilize and converge and the Monte Carlo standard errors behave in the same way as with a single chain. The density estimates are very similar across chains (as in the plot of Density Estimates with Different Starting Values in Figure 3). Hence the result is not sensative to starting values when sample size is large enough.

Figure 3. MCMC Diagnostics

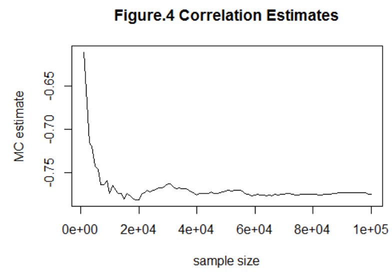


2. (a) Similar to that in Problem 1, the likelihood of $(\beta_0, \beta_1, \lambda | Y, X, \sigma_i) \propto \prod_{i=1}^{n} f(Y_i; \beta_0 + \beta_1 X_i, \sigma_i, \lambda)$ where f is the pdf for EMG.
The priors are $\beta_0 \sim N(0, 10)$, $\beta_1 \sim N(0, 10)$, $\lambda \sim Gamma(0.01, 100)$
Assuming these parameters are independent, the posterior joint density $\pi(\beta_0, \beta_1, \lambda | Y, X, \sigma_i) \propto h(\beta_0, \beta_1, \lambda) = \mathcal{L}(\beta_0, \beta_1, \lambda | Y, X, \sigma_i) \times f_{prior}(\beta_0) \times f_{prior}(\beta_1) \times f_{prior}(\lambda)$
In the following variables-at-a-time algorithm, I took the "lazy" approach of using the joint posterior distribution for the full-conditional distributions because of proportionality. And I used the same proposal density all parameters: $q(x|x(t)) \sim N(x(t), 0.5^2)$
To sample from $\pi(\beta_0, \beta_1, \lambda | Y, X, \sigma_i)$ n times:
Step 0: Start with $(\beta_0(0), \beta_1(0), \lambda(0))$, selected from a preliminary MCMC run of 1000 chains.
Repeat the following steps for t from 0 to (n-1) to get $((\beta_0(1), \beta_1(1), \lambda(1)), (\beta_0(2), \beta_1(2), \lambda(2)), ..., (\beta_0(n), \beta_1(n), \lambda(n))$:
Step 1: Update β_0:
(a). Generate a candidate β_0^* from $q(x|x(t))$
(b). Calculate acceptance probability $p = min(1, \frac{h(\beta_0^*, \beta_1(t), \lambda(t))}{h(\beta_0(t), \beta_1(t), \lambda(t))})$ ($q's$ cancell out due to symmetry)
(c). Generate a random u from unif$(0, 1)$. If $u < p$, then $\beta_0(t+1) = \beta_0^*$, else $\beta_0(t+1) = \beta_0(t)$
Step 2: Update β_1
(a). Generate a candidate β_1^* from $q(x|x(t))$
(b). Calculate acceptance probability $p = min(1, \frac{h(\beta_0(t+1), \beta_1*, \lambda(t))}{h(\beta_0(t+1), \beta_1(t), \lambda(t))})$ ($q's$ cancell out due to symmetry)
(c). Generate a random u from unif$(0, 1)$. If $u < p$, then $\beta_1(t+1) = \beta_1^*$, else $\beta_1(t+1) = \beta_1(t)$
Step 3:Update λ
(a). Generate a candidate λ^* from $q(x|x(t))$
(b). Calculate acceptance probability $p = min(1, \frac{h(\beta_0(t+1), \beta_1(t+1), \lambda^*)}{h(\beta_0(t+1), \beta_1(t+1), \lambda(t))})$ ($q's$ cancell out due to symmetry)
(c). Generate a random u from unif$(0, 1)$. If $u < p$, then $\lambda(t+1) = \lambda^*$, else $\lambda(t+1) = \lambda(t)$

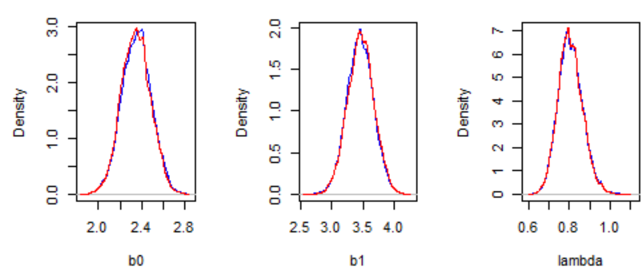
(b)

Parameter	$\hat{\mu_n}$ (MCMCse)	95% Credible Interval
β_0	2.35 (0.003)	(2.08, 2.61)
β_1	3.45 (0.004)	(3.04, 3.86)
λ	0.81 (0.001)	(0.70, 0.93)

(c) $\hat{Corr}(\beta_0, \beta_1) = -.78$. A plot of correlation estimate as chain updates can be found in Figure 4.

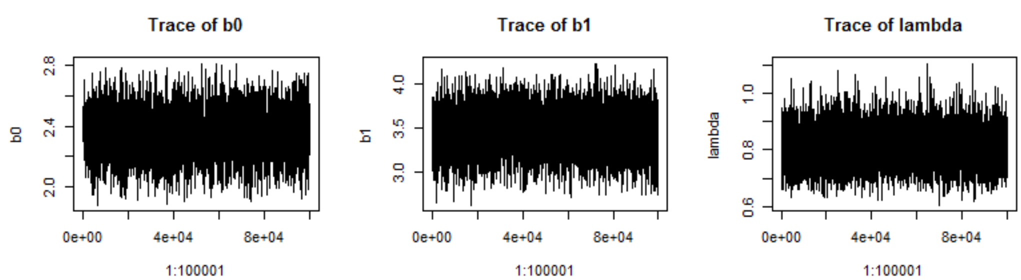


(d) Figure 5. Estimated Marginal Density Plots (Blue: after 100,000 samples; Red: after 50,000 samples)



(e) Examining the traces of β_0, β_1 and λ in the above chain, we can see that the chain is mixing faily well.

Figure 6. Trace Plots of Parameters in One Chain



The effective sample size is over 6000, which is generally good enough for mean and CI estimates. Looking at the trend of mean estimates and MCMCse in one chain (Figure 7.), mean estimates stabilize and MCMCse decrease as chain processes (though for β_0 the MCMCse is not decreasing as fast and we see more fluctuation in its mean estimate over time).

In addition, I ran this sampler with 5 sets of different starting values and obtained the following result (Figure 8 and 9). Again the mean estimates stabilize and converge, and the Monte Carlo standard errors behave in the same way as with a single chain. The density estimates are very similar across chains. Hence the approximation was not influenced much by starting values when the chain is long enough.

Figure 7. Mean Estimates and MCMCse in One Chain

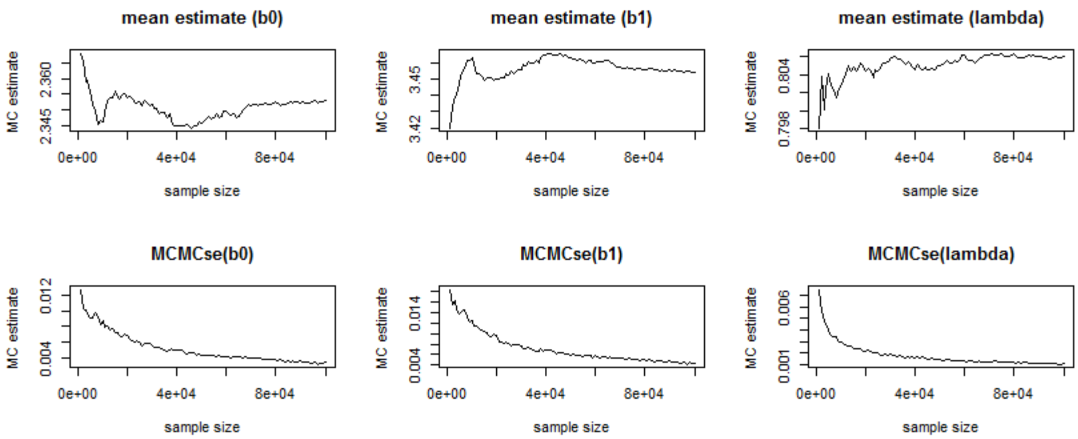


Figure 8. Mean Estimates and MCMCse across Five Chains

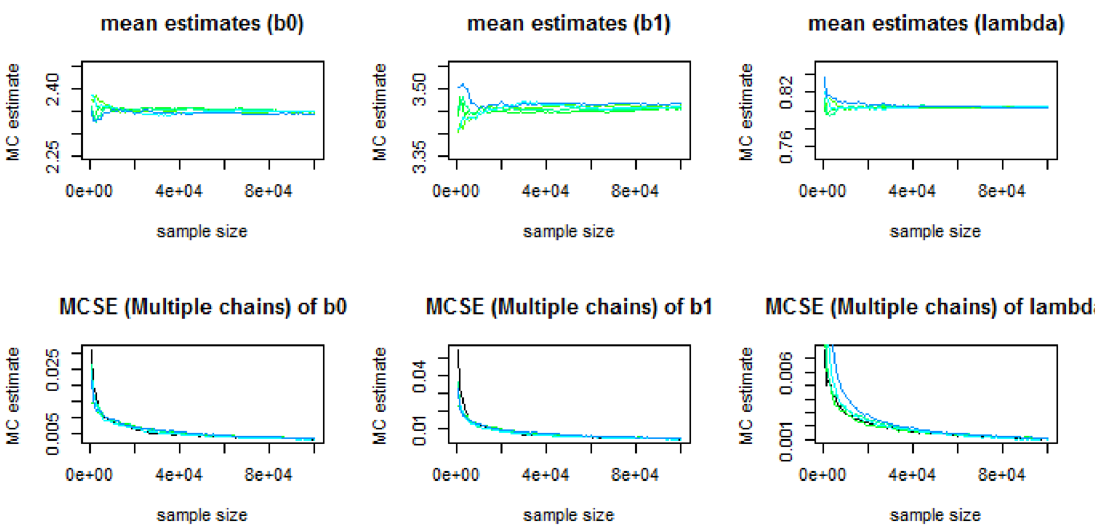
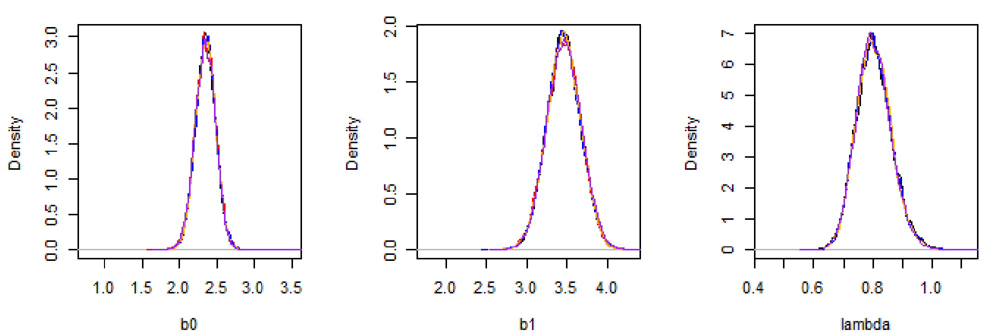


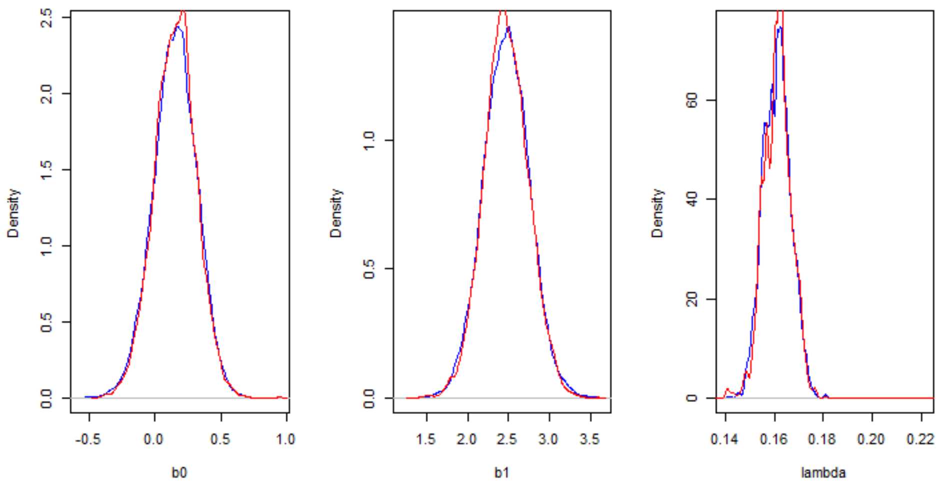
Figure 9. Density Estimates across Five Chains (Blue: After 100,000 iterations; Red: After 50,000 iterations)



3. (a)

Parameter	$\hat{\mu_n}$ (MCMCse)	95% Credible Interval
β_0	0.15 (0.004)	(-0.18, 0.47)
β_1	2.47 (0.006)	(1.92, 3.02)
λ	0.16 (0.000)	(0.15, 0.17)

(b) Figure 10. Density Estimates across Five Chains (Blue: After 100,000 iterations; Red: After 50,000 iterations)



4. When I applied the algorithm I used in Problem 2 here, I found 1) the chain wasn't mixing as fast, 2) the chain wasn't mixing as well, especially for λ. I speculate that it might be because more data resulted in a "narrower" posterior. Hence I first ran a slightly longer chain (2,500) to find a "better" starting value, and then I used proposal functions with smaller sd's. The estimates, in fact, did not change too much. (I guess it's because I have a really long chain?) However, effective sample size increased form 6,700 to 7,100. Making these changes also increased acceptance rates (from (20%, 33%, 1%) to (25%, 40%, 3%)). The mixing of λ seems to have improved based on the trace plot (Figure 11 upper), and the autocorrelation of λ decreased in comparison (Figure 11 lower).

Figure 11. Comparion between Two Algorithms