Problem 1

(a)

The Metropolis-Hastings algorithm:

1. Starting value for MCMC Chain, $\beta_1^{(0)} = 5$, was chosen based on preliminary MCMC runs.
2. Update the variable with the proposal function. This requires a Metropolis-Hastings update:
   (a) Propose a new value for β_1, β_1^*, according to the proposal distribution. Note that $\tau = 1$ for this random walk Normal distribution proposal, based on preliminary runs and diagnostics.
   $$q(\beta_1|\beta_0, \sigma_i, \lambda, \mathbf{Y}) \sim \mathbf{Norm}(\beta_\mathbf{1}^{\mathbf{i-1}}, \mathbf{1})$$
   .
   (b) Compute the Metropolis-Hastings accept-reject ratio. Because log densities are used, this fraction becomes the exponential of the two posteriors subtracted.
   $$\alpha(\beta_1, \beta_1^*) = \min\left(\frac{f(\beta_1^*|\beta_0, \sigma_i, \lambda, \mathbf{Y})q(\beta_1|\beta_0, \sigma_i, \lambda, \mathbf{Y})}{f(\beta_1|\beta_0, \sigma_i, \lambda, \mathbf{Y})q(\beta_1^*|\beta_0, \sigma_i, \lambda, \mathbf{Y})}, 1\right)$$
   (c) Accept the new value β_1^* with probability $\alpha(\beta_1, \beta_1^*)$, otherwise reject β_1^*.
3. These values now constitute a new Markov chain state.
4. Return to step 2 N-1 times to produce a Markov chain of length N.

(b)

Estimate = 5.810 MCMC st err = 0.009

(c)

95% CI: (5.043, 6.511)

(d)

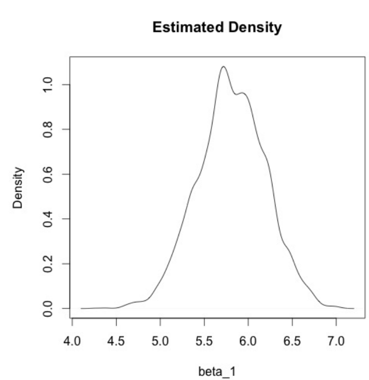

Figure 1. Plot of estimate of the posterior pdf of β_1.

Problem 1 continued on next page...

(e)

The main variation implemented to determine that the approximations above are appropriate was different τ values for the proposal function. In an effort to control these τ values for subsequent chains, the MCMC algorithm was built with τ as one if its parameters. This change increases the time each chain runs, but it was determined to be beneficial over coding multiple proposal function to alternate between.

Diagnostic tools included the use of estimates vs sample size plot, autocorrelation plots, running mean plots, trace plots of the MCMC chain, and density plots (see Figure 2). It was determined through these diagnostics that a proposal function with $\tau = 1$would be utilized for the final approximations. Additionally, MCMC standard error (0.008) and effective sample size (2177.1 - higher than all other chains considered) were considered and found to be acceptable. As can be seen in Figure 2, $\tau = 1$ provides the MCMC chain with relatively minimal autocorrelation, consistent running mean, and acceptable amount of mixing when compared with other values of τ. It should be noted that preliminary MCMC chains included a wider range of values for τ, however these results focus on a smaller range about the final value used.

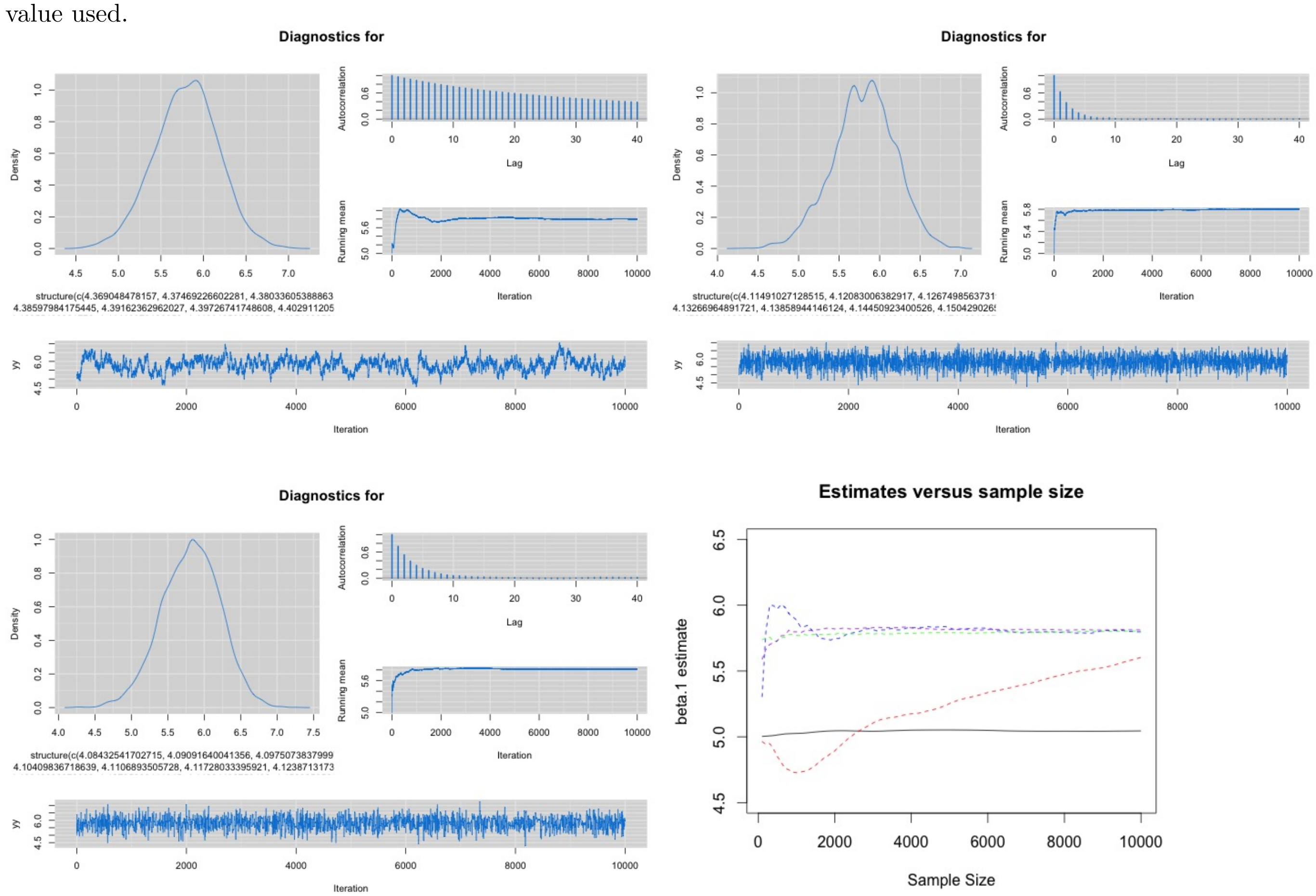

Figure 2. Three diagnostic plots for estimating β_1 with differing τ values used in proposal function. Top left: $\tau = .1$. Top Right: $\tau = 1$. Bottom left: $\tau = 2$ Bottom right: Estimates of β_1 versus sample size. Black - $\tau = 0.001$, red - $\tau = 0.01$, blue - $\tau = 0.1$, green - $\tau = 1$, and purple - $\tau = 2$.

Problem 2

(a)

The Metropolis-Hastings algorithm:

1. Pick starting values for the Markov chain. The values $(\beta_0^{(0)}, \beta_1^{(0)}, \lambda^{(0)}) = (2, 3, 1)$ were chosen based on preliminary MCMC runs.
2. These all require a Metropolis-Hastings update:
   (a) Propose β_0, β_1, and λ from (random walk Normal distribution) proposal distribution for those values
   $$(\beta_0^{(i)}, \beta_1^{(i)}), \lambda^{(i)}) \sim N((\beta_0^{(i-1)}, \beta_1^{(i-1)}), \lambda^{(i-1)}), (0.1, 0.1, 0.1))$$
   (b) Compute the Metropolis-Hastings accept-reject ratio,
   $$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta^*}) = min\left(\frac{f(\boldsymbol{\theta^*}|\sigma_i, \mathbf{Y})q(\boldsymbol{\theta}|\sigma_i, \mathbf{Y})}{f(\boldsymbol{\theta}|\sigma_i, \mathbf{Y})q(\boldsymbol{\theta^*}|\sigma_i, \mathbf{Y})}, 1\right)$$
   (c) Accept the new values $\boldsymbol{\theta^*}$ with probability $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta^*})$, otherwise reject $\boldsymbol{\theta^*}$, i.e., the next values of $\boldsymbol{\theta}$ remains the same as before.
3. These new values (either proposed or repeated) now constitute a new Markov chain state.
4. Return to step 2 N-1 times to produce a Markov chain of length N.

(b)

Table 1: Table of Relevant Information for Each Parameter

Parameter	Posterior Mean (MCMC St Err)	Posterior 95% CI
β_0	2.388 (0.009)	(2.134, 2.643)
β_1	3.456 (0.014)	(3.059, 3.863)
λ	0.807 (0.002)	(0.694, 0.928)

(c)

Correlation: -0.756

(d)

See part (e).

(e)

This algorithm was found to be a reliable source for estimations of the three parameters in a very similar manner from Question 1. Several values for τ were considered for the proposal function. Unlike Question 1 part e, for the sake of space I will only be presenting plots and statistics on the final algorithm used in an effort to justify to produce estimates reported above. The exception to this is the plots of parameter estimates vs sample size. Figure 3 shows that while medium-high levels of autocorrelation are present, mixing is at an acceptable level so thinning was not used.

Problem 2 [(e)] continued on next page...

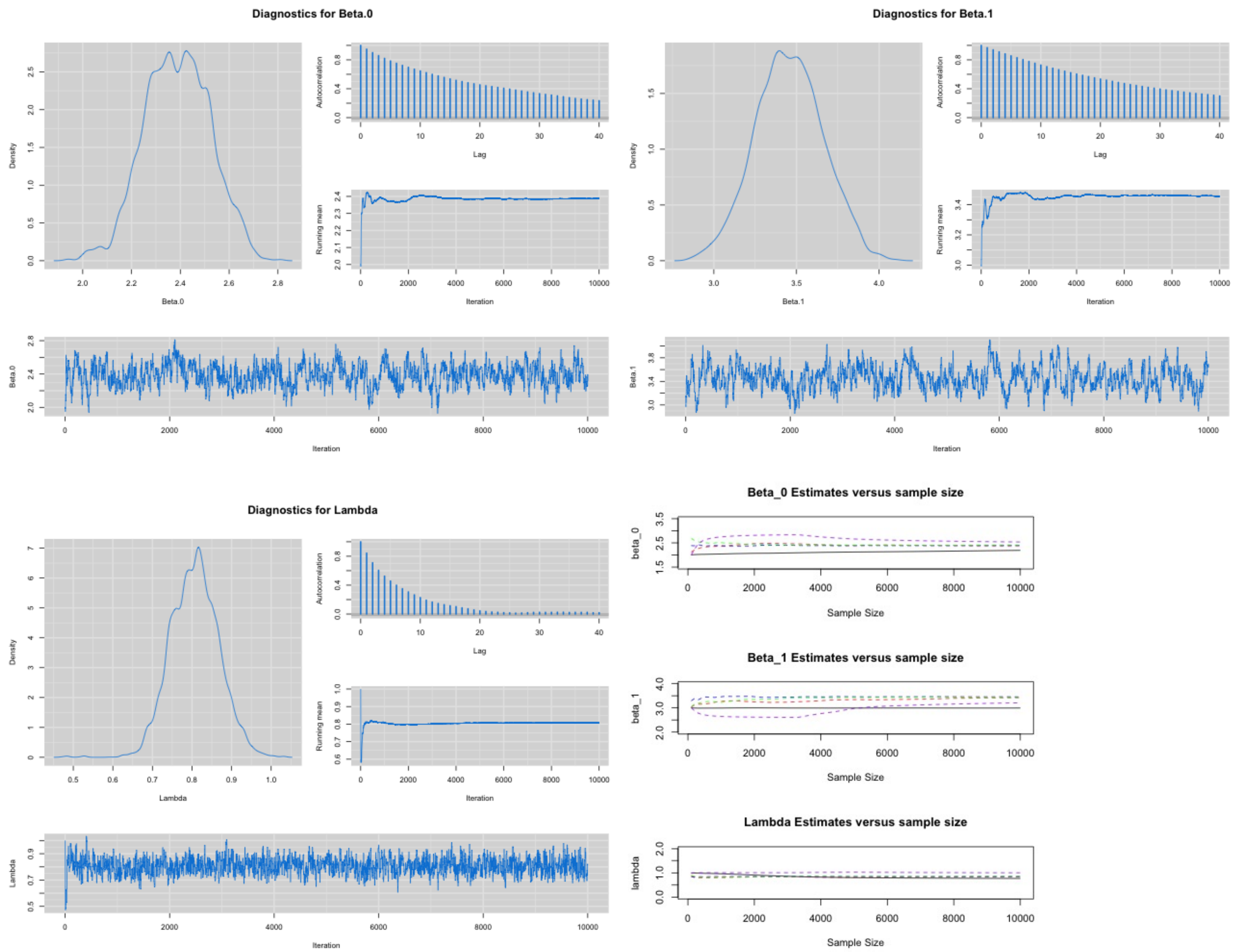

Figure 3. Diagnostic plots used to justify the use of $\tau = 0.1$. Top left: β_0. Top Right: β_1. Bottom left: λ Bottom right: Estimates of β_0, β_1, and λ versus sample size. Black - $\tau = 0.001$, red - $\tau = 0.01$, blue - $\tau = 0.1$, green - $\tau = 1$, and purple - $\tau = 2$.

Problem 3

(a)

Table 2: Table of Relevant Information for Each Parameter

Parameter	Posterior Mean (MCMC St Err)	Posterior 95% CI
β_0	0.178 (0.011)	(-0.125, 0.497)
β_1	2.427 (0.020)	(1.881, 2.936)
λ	0.161 (0.0001)	(0.150, 0.173)

(b)

See part (c).

(c)

One major difference between this data set and the data set for Question 2 is the relatively higher autocorrelation for the β_0 and β_1 estimations. Now, thinning (using only every nth step) is a proposed fixed for high level autocorrelation. However, thinning is not usually appropriate when the goal is precision of estimates from an MCMC sample. Sacrificing variance and information is not the best route in this scenario (although if memory or time constraints were prioritized higher than thinning may be a more viable option).
Ultimately, the same code was used for both Questions 2 and 3. Figure 4 shows that while high levels of autocorrelation are present, the estimate means still converge nicely in the time alloted (20000 iterations).

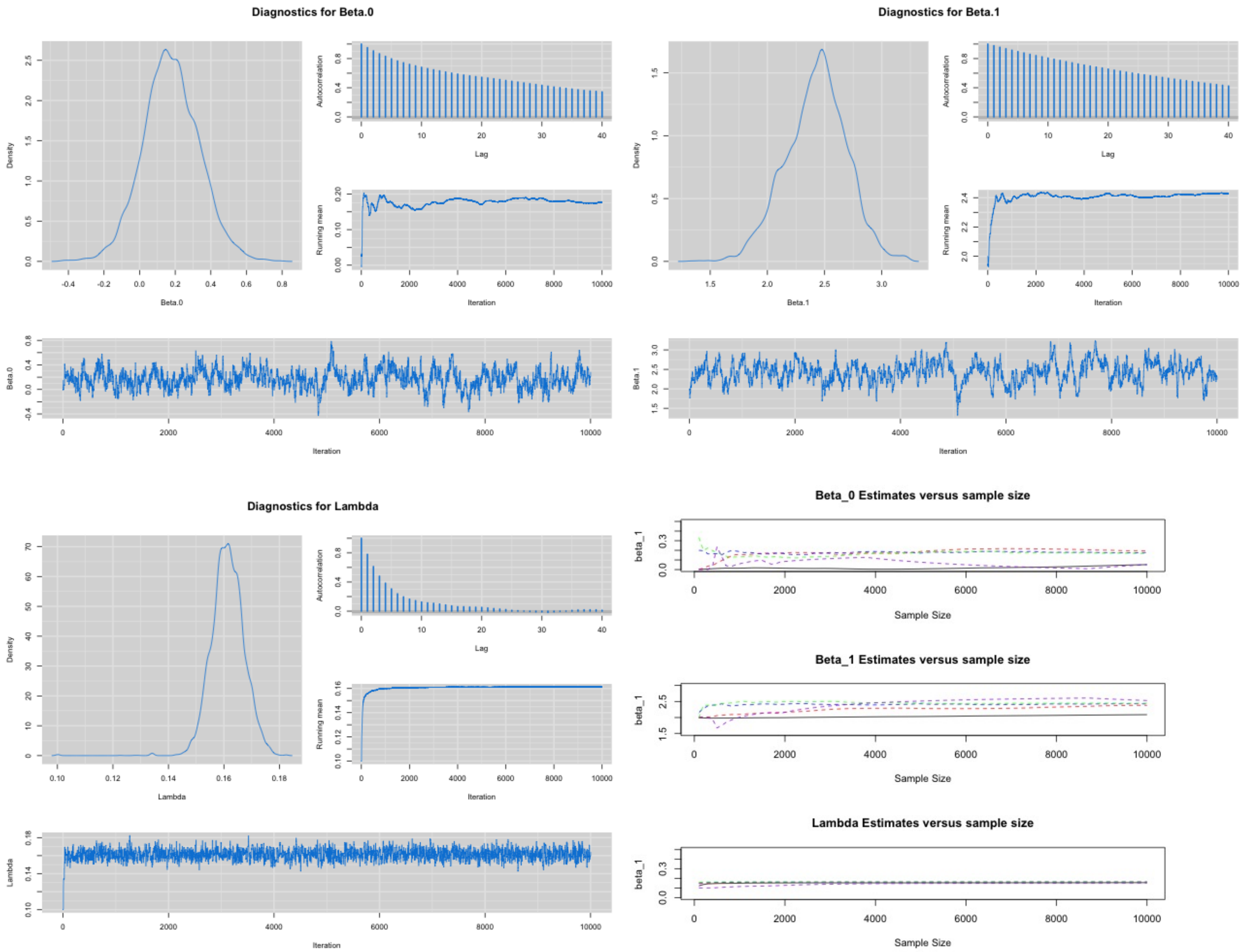

Figure 4. Diagnostic plots used to justify the use of $\tau = 0.1$ and no thinning. Top left: β_0. Top Right: β_1. Bottom left: λ Bottom right: Estimates of β_0, β_1, and λ versus sample size. Black - $\tau = 0.001$, red - $\tau = 0.01$, blue - $\tau = 0.1$, green - $\tau = 1$, and purple - $\tau = 2$.