1

(a)

First I do some preliminary calculations.

$$
\begin{aligned}
L_i(\beta_1) &= f(y_i; 0.5 + \beta_1 x_i, 1, 0.4) \\
l_i(\beta_1) &= \log[f(y_i; 0.5 + \beta_1 x_i, 1, 0.4)] \\
L(\beta_1) &= \prod_{i=1}^{n} L_i(\beta_1) \\
p(\beta_1) &\propto \exp\left(-\frac{{\beta_1}^2}{200}\right) \\
\pi(\beta_1) &\propto L(\beta_1) \cdot p(\beta_1) \\
&\propto \exp\left(-\frac{{\beta_1}^2}{200}\right) \cdot \prod_{i=1}^{n} L_i(\beta_1) \\
&= h(\beta_1) \\
\log[h(\beta_1)] &= -\frac{{\beta_1}^2}{200} + \sum_{i=1}^{n} l_i(\beta_1)
\end{aligned}
$$

I chose the initial values of the chain after running the chain a few times and seeing how it was moving. From the initial value we generate a new point by using a normal random number generator with the mean as the previous value and some standard deviation that will be changed as appropriate. For every step, we either accept or reject the proposed new point by taking a log of the uniform random variable from 0 to 1 and comparing it with the value of the difference of the last equation shown above evaluated at the new point from the last equation shown above evaluated at the previous point. Note that if this difference is larger than 0, we will always accept as the log of a number from 0 to 1 is always less than 0 (i.e. there is no need to take a min although I did it anyways). I did 10,000 steps for the chain. This was also the case for 2 and 3. The acceptance rate was 0.4808. The tuning parameter was 0.65 and the initial calue was 7.

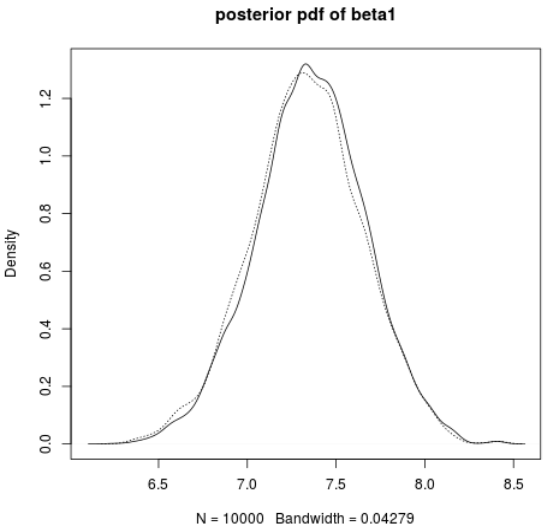
(b)

Posterior expectation: 7.357395
MCMC standard error: 0.006749956
7.357395 ± 0.01322991 (95%)

(c)

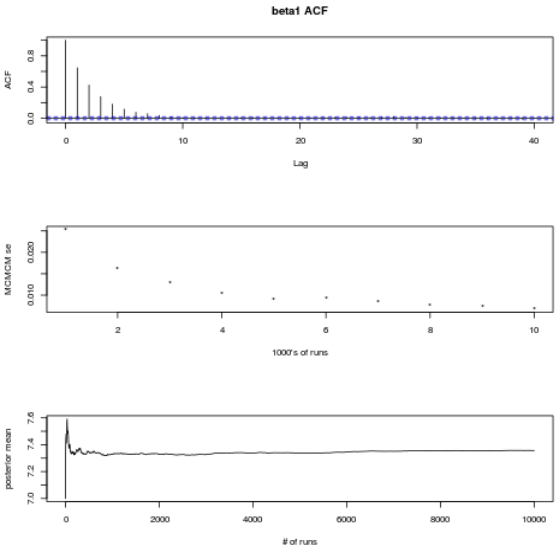
(6.741909,7.961197)

(d)



(e)

The autocorrelation did not decay as fast as expected and the ESS was 2126.792. Even though tests with 100 runs gave ESS of values larger than 50. As can be seen in the diagram below, the MCMC standard error was plateauing out and the mean was converging so this was significant reason to believe this approximation was good.



Furthermore, in the plot in (d) I showed the density after 5000 runs in a dotted line. From this we can also see that the density was already reaching quiet a stable state. Attempts in small runs with odd initial conditions quickly resulted in the chain being pulled toward the mean.

2

(a)

$$
\begin{aligned}
L_i(\beta_0,\beta_1,\lambda) &= f(y_i;\beta_0+\beta_1 x_i,1,\lambda)\\
l_i(\beta_0,\beta_1,\lambda) &= \log[f(y_i;\beta_0+\beta_1 x_i,1,\lambda)]\\
L(\beta_0,\beta_1,\lambda) &= \prod_{i=1}^{n} L_i(\beta_0,\beta_1,\lambda)\\
p(\beta_0) &\propto \exp\left(-\frac{{\beta_0}^2}{200}\right)\\
p(\beta_1) &\propto \exp\left(-\frac{{\beta_1}^2}{200}\right)\\
p(\lambda) &\propto \lambda^{-0.99}\exp -\frac{\lambda}{100}\\
\pi(\beta_0,\beta_1,\lambda) &\propto L(\beta_0,\beta_1,\lambda)\cdot p(\beta_0)\cdot p(\beta_1)\cdot p(\lambda)\\
&\propto \exp\left(-\frac{\beta_0^2+{\beta_1}^2+2\lambda}{200}\right)\lambda^{-0.99}\\
&\quad\cdot\prod_{i=1}^{n} L_i(\beta_0,\beta_1,\lambda)\\
&= h(\beta_0,\beta_1,\lambda)\\
\log[h(\beta_0,\beta_1,\lambda)] &= -\frac{\beta_0^2+{\beta_1}^2+2\lambda}{200}-0.99\log(\lambda)\\
&\quad+\sum_{i=1}^{n} l_i(\beta_0,\beta_1,\lambda)\\
\log[h(\beta_0,\beta_1,\lambda)] &= -\frac{\beta_0^2+{\beta_1}^2+2\lambda}{200}-0.99\log(\lambda)\\
&\quad+\sum_{i=1}^{n} l_i(\beta_0,\beta_1,\lambda)\\
\log[h(\beta_0|\beta_1,\lambda)] &= -\frac{\beta_0^2}{200}+\sum_{i=1}^{n} l_i(\beta_0,\beta_1,\lambda)\\
\log[h(\beta_1|\beta_0,\lambda)] &= -\frac{\beta_1^2}{200}+\sum_{i=1}^{n} l_i(\beta_0,\beta_1,\lambda)\\
\log[h(\lambda|\beta_0,\beta_1)] &= -\frac{\lambda}{100}-0.99\log(\lambda)+\sum_{i=1}^{n} l_i(\beta_0,\beta_1,\lambda)
\end{aligned}
$$

I was initially trying to do variable at a rime metropolis hasting but I ended up updateing β_0 and β_1 toghether as a block, in light of the fact that they are correlated. So I used the equation below as well as the last equation shown above.

$$\log[h(\beta_1,\beta_0|\lambda)] = -\frac{\beta_0^2}{200}-\frac{\beta_1^2}{200}+\sum_{i=1}^{n} l_i(\beta_0,\beta_1,\lambda)$$

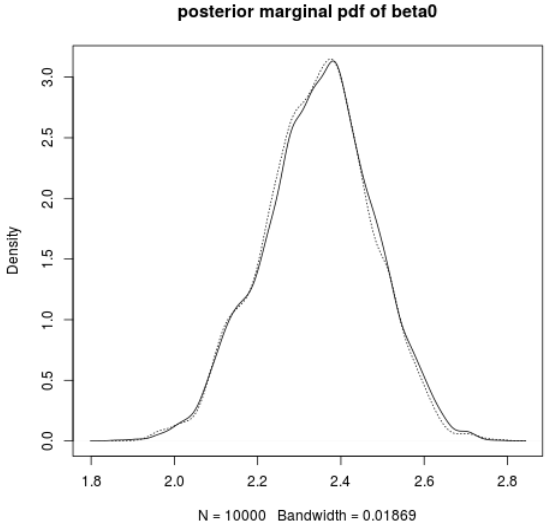
I once again, chose the initial values of the chain after running the chain a few times and seeing how it was moving. In each loop, for β_0 and β_1, we start from the initial value to generate two new independent points by using two normal random number generator with the mean as the previous value. We choose some standard deviation for the normals that will be changed as appropriate. For every step, we either accept or reject these two proposed new points by taking a log of the uniform random variable from 0 to 1 and comparing it with the value of the difference of $\log[h(\beta_1,\beta_0|\lambda)]$ evaluated at the new points from $\log[h(\beta_1,\beta_0|\lambda)]$ evaluated at the previous points. Then we generate another new point for the new λ in the loop. Since λ is positive we truncate the normal so it is only supported on the positive real line. We do this by repeatedlt sampling from a normal until it is positive. This means that $q(y,x)\neq q(x,y)$ where y is the new point and x is the old point, since the area of the truncated region will differ when the mean of the normal random sample changes. But since the only difference is the normalizing constant, we simply take q to be the integral of the truncated normal. So the difference of the log of $q(y,x)$ and $q(x,y)$ is added to the difference of the log of $\log[h(\lambda|\beta_0,\beta_1)]$ and compared with the log of an uniform just like before. Note, when calculating these quantities we always use the newest β_0 and β_1. The tuning parameters were 0.1, 0.15, 0.05 and the initial values were 2.5, 3.3, 1.

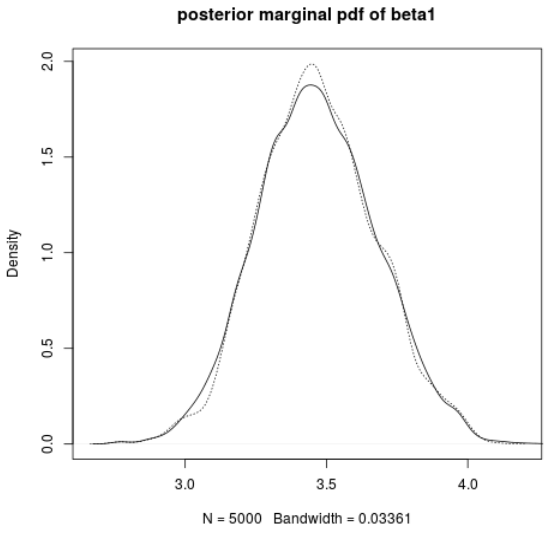
(b)

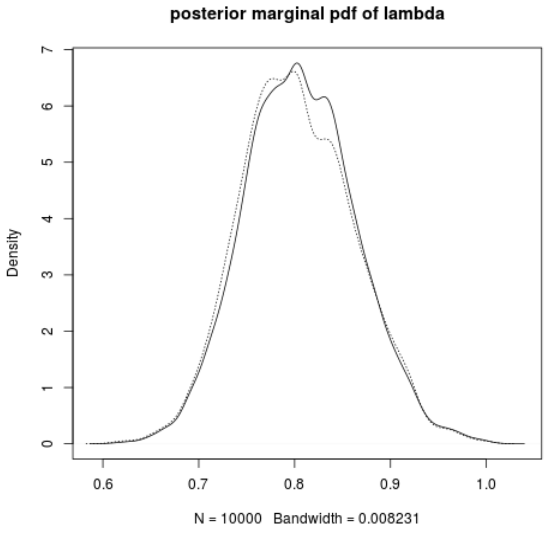
	mean (95% errors)	95% credible interval
β_0	2.34645 ± 0.01236037	(2.081949,2.601203)
β_1	3.467246 ± 0.02051742	(3.069941,3.887433)
λ	$0.8060581 \pm 0.004105278$	(0.6979463,0.9219136)

(c)
-0.7684939

(d)

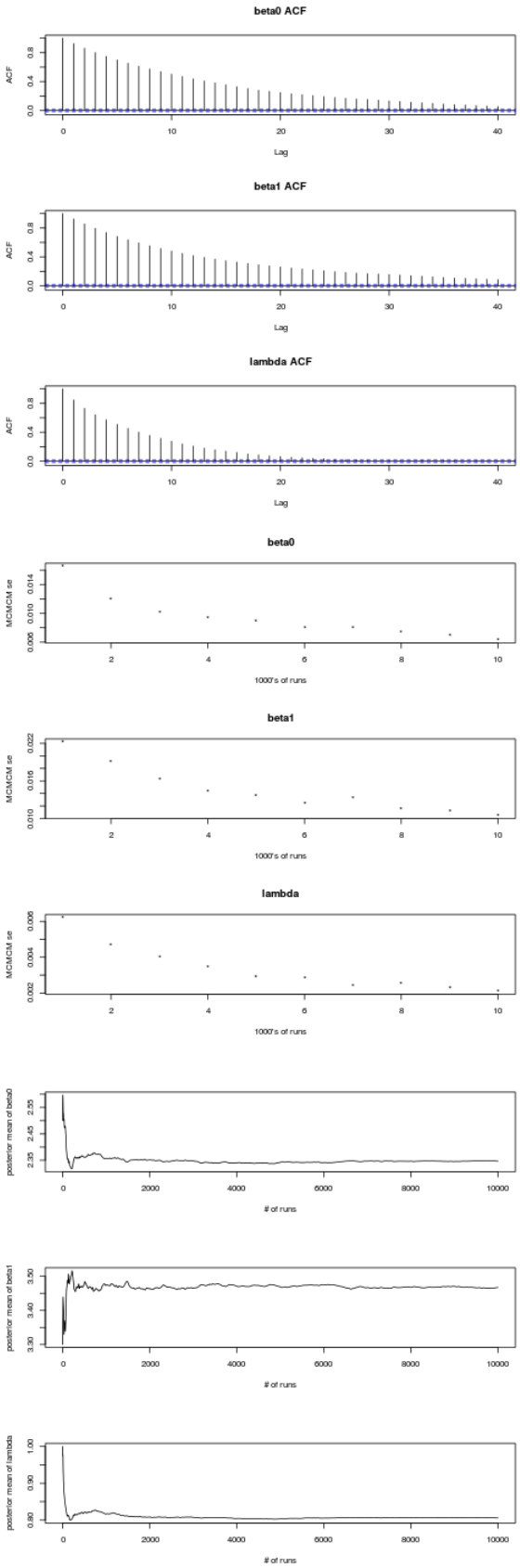






(e)

The autocorrelation did not decay as fast as desired and the ESS for β_0, β_1, λ was, respectively, 370.9955, 364.3535, 692.5058. For β_0 and β_1, actual ESS values may be even lower as the autocorrelation for lags larger than what can be reliably calculated may also be significant. But the convergence of the mean is clear in all three cases and that is a good sign.



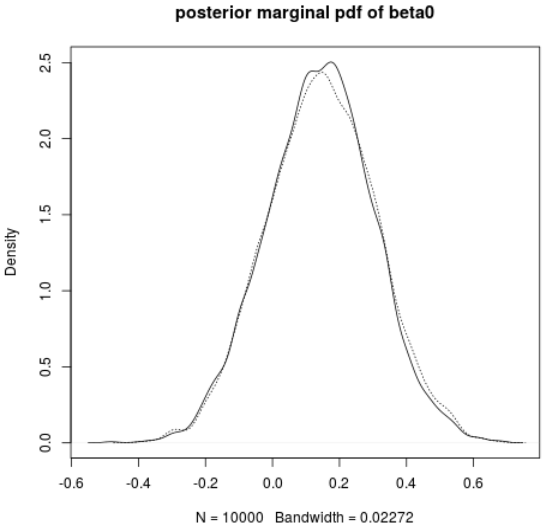
Furthermore, if you look at the density after 5000 runs shown as dotted lines in (d), the density can be said to have reached stability after 5000 runs. This is another reason to believe the approximation is acceptable. It is likely that we can further lower the MCMC standard error but it is still prertty good. Attempts in small runs with odd initial conditions quickly resulted in the chain being pulled toward the mean. The acceptance rate was 0.4485 and 0.6759.

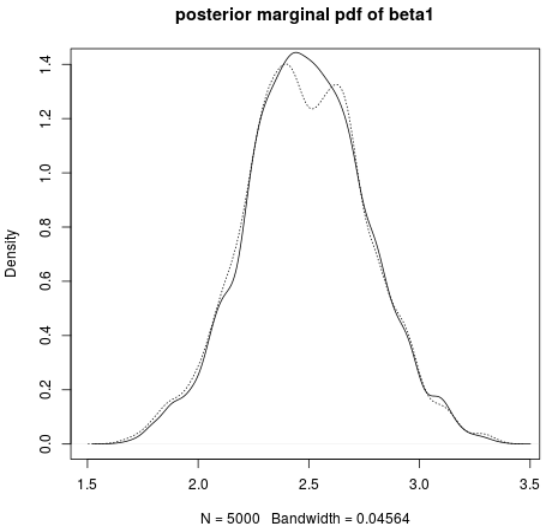
3

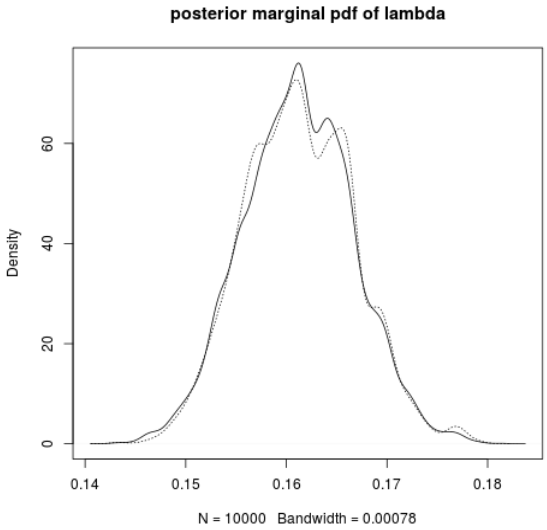
(a) The tuning parameters were 0.2, 0.2, 0.05 and the initial calues were 0, 2.8, 0.15.

	mean (95% errors)	95% credible interval
β_0	0.1417015 ± 0.01595711	$(-0.1712379, 0.4577729)$
β_1	2.492116 ± 0.0293749	$(1.956825, 3.050589)$
λ	$0.1611844 \pm 0.000389325$	$(0.1506172, 0.1720221)$

(b)

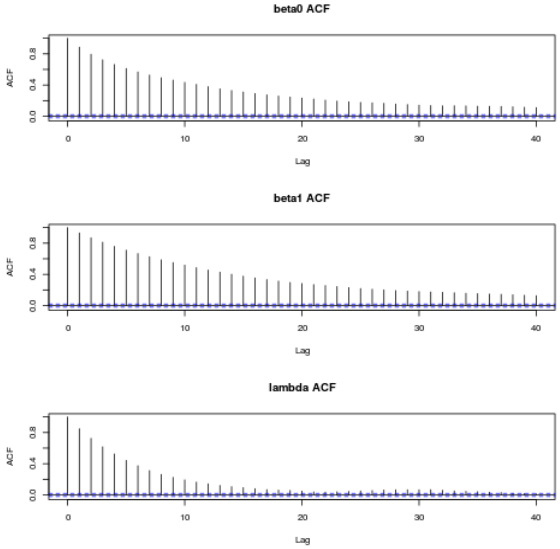


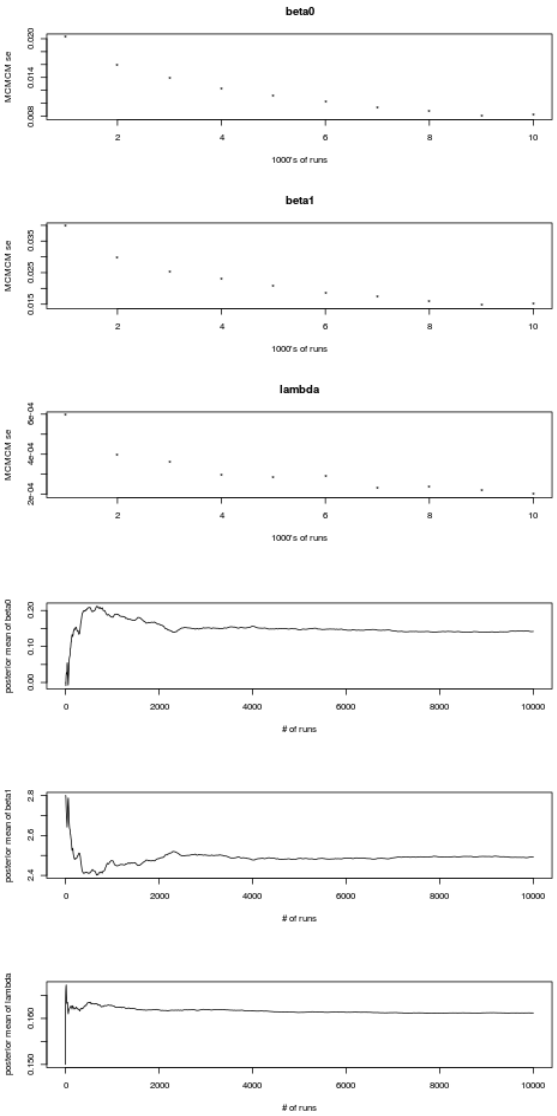




(c)

I modified the tuning parameters through a process of trial and error.





The autocorrelation and ESS does not look good, but the convergence of the mean is clear and the densities after 5000 runs shown in a dotted line seems to be stable. It is likely that we can further lower the MCMC standard error but it is still prertty good. Attempts in small runs with odd initial conditions quickly resulted in the chain being pulled toward the mean. The acceptance rate was 0.3633 and 0.1326.