

STAT 515
Homework #11 WITH SOLUTIONS

1. Suppose that Y_1, \dots, Y_n is a simple random sample from $N(\theta, 1)$, where we assume that $(\log \theta - \mu)/\sigma$ has a t distribution on r degrees of freedom (this is the example we discussed briefly in class on April 11). Assume that μ , σ , and r are known (parameters on the prior distribution like this are called *hyperparameters* in a Bayesian context).
- (a) Describe a Metropolis-Hastings algorithm for sampling from the posterior distribution of $\theta \mid \mathbf{Y}$, using a normal proposal distribution centered at the current value of the Markov chain and with variance τ^2 .

Solution: Ignoring multiplicative constants, the likelihood (the joint density for Y_1, \dots, Y_n) times the prior for θ equals

$$\frac{1}{\theta} \left[1 + \frac{1}{r} \left(\frac{\log \theta - \mu}{\sigma} \right)^2 \right]^{-(r+1)/2} \exp \left\{ -\frac{1}{2} \sum_{i=1}^n (Y_i - \theta)^2 \right\} I\{\theta > 0\}.$$

Let us call this function $\pi(\theta)$ to simplify notation. Letting θ^* denote a proposed next value and θ the current value of the chain, we conclude that a normal proposal distribution as specified leads to a Metropolis-Hastings acceptance ratio equal to $\pi(\theta^*)/\pi(\theta)$, since the normal proposal density is symmetric (that is, $q(\theta \mid \theta^*) = q(\theta^* \mid \theta)$ if we take $q(x \mid y)$ to be a normal density function with mean y evaluated at x).

We conclude that the Metropolis-Hastings algorithm consists of first simulating a normal random value θ^* with mean θ and variance τ^2 , then accepting θ^* (replacing θ by θ^*) as long as a standard uniform random variable is less than $\pi(\theta^*)/\pi(\theta)$.

However, the algorithm should calculate the ratio on the log scale for the sake of numerical stability. This means that we accept θ^* whenever

$$\log U < \log \pi(\theta^*) - \log \pi(\theta).$$

Whenever $\theta^* \leq 0$, the value of $\pi(\theta^*)$ equals zero, which means that we never accept θ^* in that case.

- (b) Take $\mu = 0$, $\sigma = 5$, and $r = 4$. For the dataset Y_1, \dots, Y_{100} at <http://sites.stat.psu.edu/~dhunter/515/hw/hw11prob1b.txt>, implement your M-H algorithm starting the chain at $\theta_0 = 1$ and running for 50,000 steps. Use $\tau^2 = 1$.

Solution: In R:

```
> # First, we create a function to calculate log(pi):
> logpi <- function(x, data, mu=0, sigma=5, r=4) {
+   if (x<0)
+     return (-Inf)
+   else
+     return ( -(r+1)/2*log(1 + ((log(x) - mu)/sigma)^2/r) - log(x) - sum((data-x)^2)/2)
+ }
> # Here is a function to return a sample from the posterior:
> samplePosterior <- function(tau=1, n=50000) {
+   y <- scan("http://sites.stat.psu.edu/~dhunter/515/hw/hw11prob1b.txt")
+   theta <- rep(1, 50001) # This will hold the theta values
+   accepts <- 0 # Keep track of the number of acceptances
+   # Here is the main loop:
+   for(i in 1:50000) {
+     thetaStar <- rnorm(1, mean=theta[i], sd=tau)
+     u <- runif(1)
+     if (thetaStar>0 && log(u) < logpi(thetaStar, y) - logpi(theta[i], y)) {
+       accepts <- accepts + 1
+     }
+   }
+ }
```

```

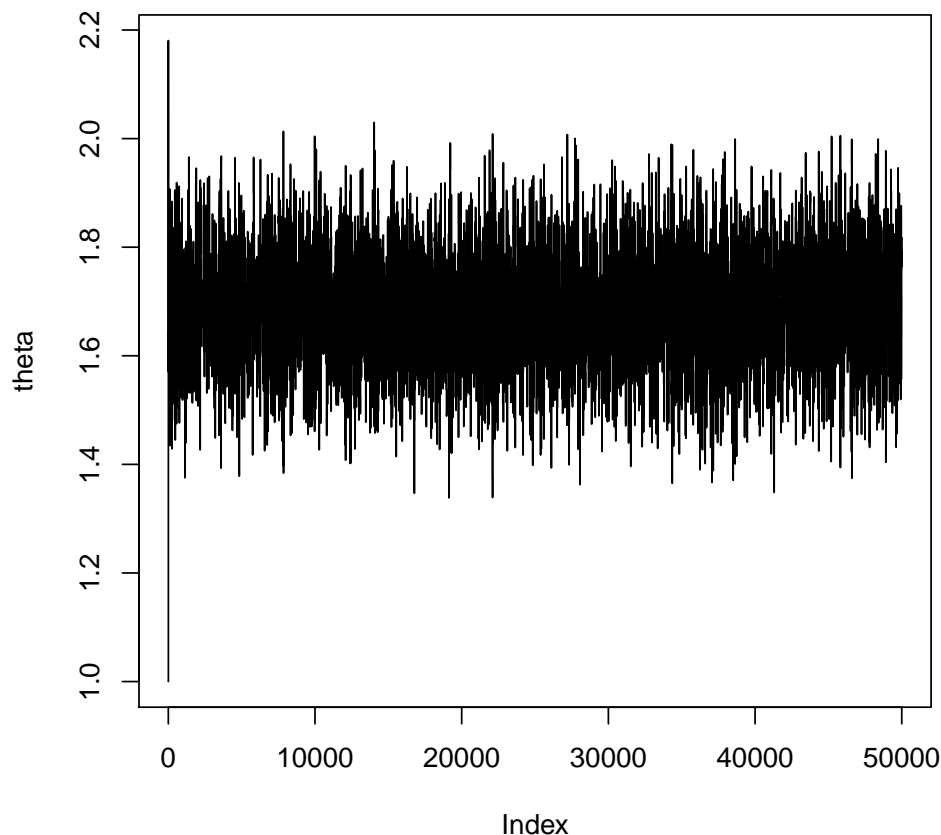
+       theta[i+1] <- thetaStar
+     } else {
+       theta[i+1] <- theta[i]
+     }
+   }
+   cat("Acceptance rate: ", accepts / n)
+   return(theta)
+ }
> theta <- samplePosterior(tau=1)
Acceptance rate: 0.12504

```

- (c) Record the acceptance rate of your MH algorithm. Then create a trace plot in which you plot the values of θ_i against i . Comment on it: Does it appear that your Markov chain is effectively “mixing”?

Solution: The acceptance rate was reported above; it is the proportion of proposals that were accepted. Here is the trace plot:

```
> plot(theta, type="l")
```

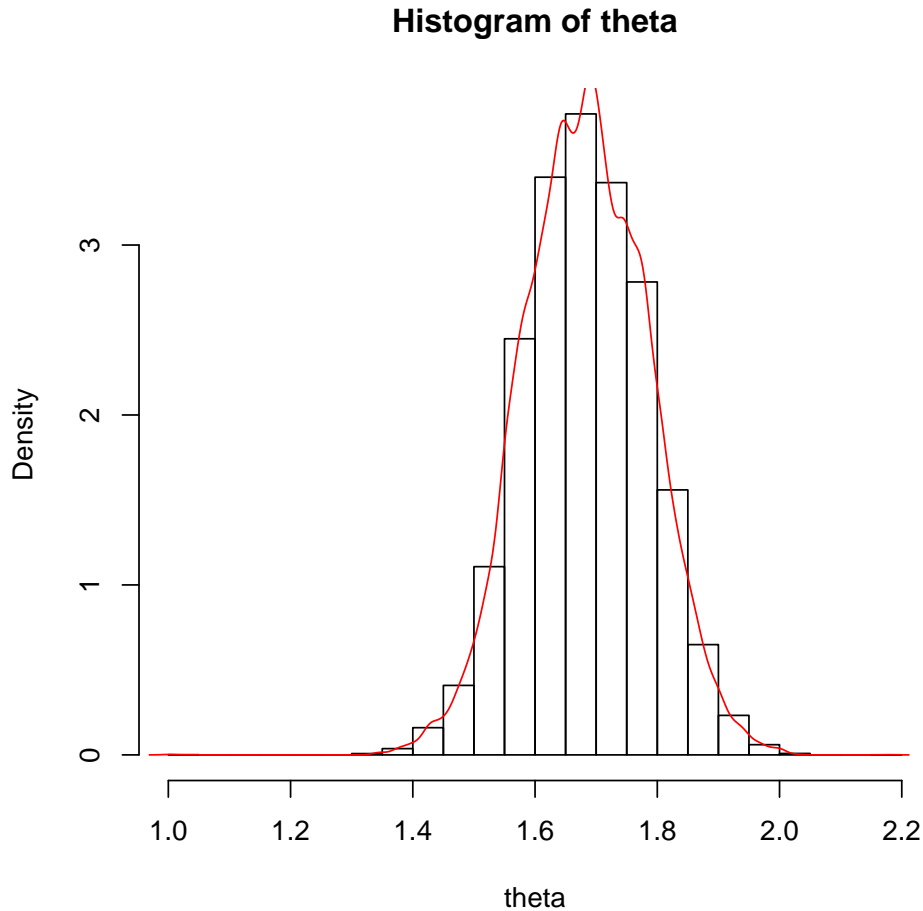


This trace plot looks great. There is no reason to think that this chain is failing to mix well based on the plot.

- (d) Create a histogram of the θ_i values. Also, report a point estimate (the posterior mean) along with a 95% credible interval for θ based on your posterior sample. (For the latter, just use the 0.025 and 0.975 sample quantiles of your sample.)

Solution: Here is a histogram (with a density estimate superimposed):

```
> hist(theta, freq=FALSE)
> lines(density(theta), col=2)
```



The posterior mean and credible interval consisting of the 0.025 and 0.975 quantiles are as follows:

```
> mean(theta)
[1] 1.683853
> quantile(theta, c(.025, .975))
      2.5%      97.5%
1.491092 1.877253
```

This credible interval reflects the uncertainty about the value of θ intrinsic to the problem. It does not reflect the uncertainty about the posterior distribution of θ due to the fact that we're using MCMC to sample from that distribution.

(e) To create a 95% confidence interval for the true posterior mean, a naive idea would be to try

$$\hat{\mu} \pm \frac{1.96}{\sqrt{n}} * \sqrt{\frac{1}{n-1} \sum_{i=1}^{50,000} (\theta_i - \hat{\mu})^2},$$

where $\hat{\mu}$ is your point estimate from part (d). Explain why this interval has a totally different interpretation than your interval from part (d). (Hint: Only one of these intervals tries to capture

the MCMC error.) Also, explain why this idea is likely to produce an interval that is narrower than it should be—and hence, it is not a very good idea from a statistical point of view.

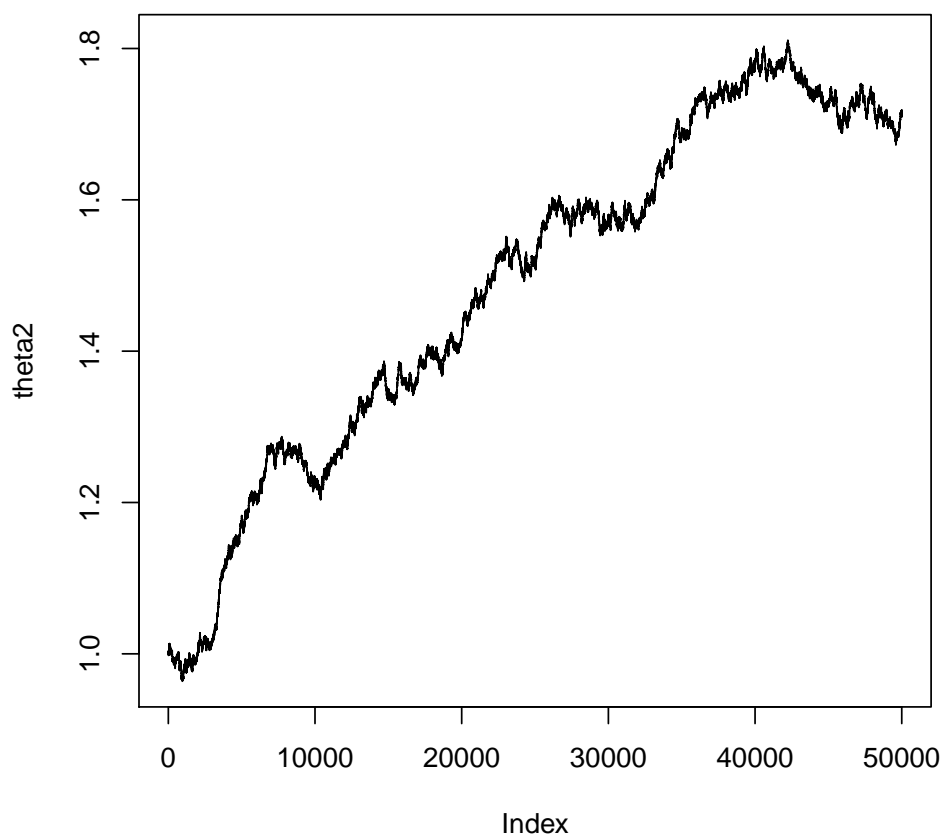
Solution: This interval only reflects uncertainty about the true posterior mean based on *this particular sample*. In other words, the posterior mean is a fixed but unknown constant, and we are using a random procedure to try to estimate it, and the confidence interval reflects our uncertainty about that fixed value due to the randomness of the procedure. It does *not* reflect the uncertainty about θ due to the fact that the Y_i were randomly chosen to begin with. (The credible interval in part (d) does that.)

The reason that the confidence interval is too narrow is that the variance of the sample mean will include some positive covariances, so it will be larger than the variance based on an i.i.d. sample from the posterior would be. Yet the naive formula uses the i.i.d. version.

- (f) Replicate part (c) using a value of τ^2 that appears “too small”. Then do the same thing for a value of τ^2 that appears “too big”. In each case, try to explain why the chain does not appear to be as effective as in part (c).

Solution: Let us try $\tau^2 = 10^{-6}$ and look at a trace plot:

```
> theta2 <- samplePosterior(tau=1/1000)
Acceptance rate: 0.99118
> plot(theta2, type="l")
```



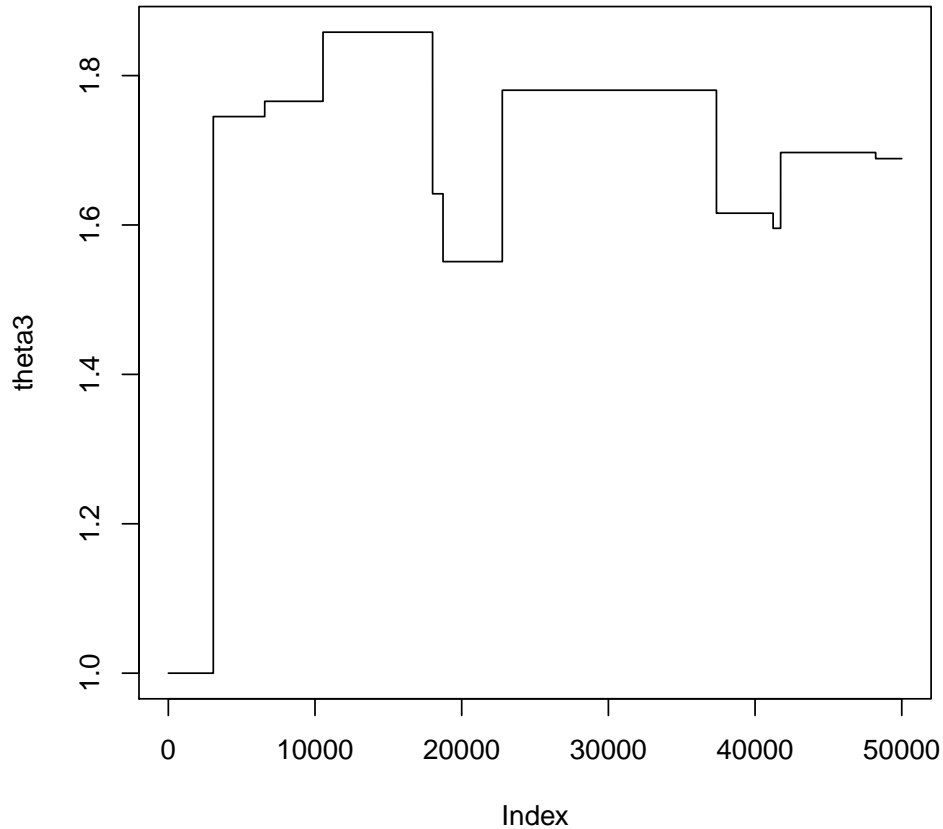
Nearly every proposal is accepted, but the traceplot reveals that mixing is very slow. This is because each proposed θ^* is very close to the current θ .

Next, consider $\tau^2 = 10^6$:

```

> theta3 <- samplePosterior(tau=1000)
Acceptance rate: 2e-04
> plot(theta3, type="l")

```



In this case, the mixing is very slow because almost none of the proposals are accepted. The reason for this is that they are very far away from θ , and the acceptance probability is therefore quite low most of the time.

2. The goal of this problem is to analyze a dataset in which the observations are assumed to come from a piecewise-homogeneous Poisson process in which the Poisson rate starts at one value and then changes at some unknown time to a different value.

The data are at <http://sites.stat.psu.edu/~dhunter/515/hw/hw11prob2.txt>, where the events have been binned into 50 time periods of equal length. A model, adapted by Murali Haran from Chapter 5 of "Bayes and Empirical Bayes Methods for Data Analysis" by Carlin and Louis (2000), for the binned counts Y_1, \dots, Y_{50} is as follows:

$$Y_i \mid k, \theta, \lambda \sim \begin{cases} \text{Poisson}(\theta) & \text{for } i = 1, \dots, k; \\ \text{Poisson}(\lambda) & \text{for } i = k + 1, \dots, 50. \end{cases}$$

The prior distributions on the k , θ , and λ parameters, along with two hyperparameters b_1 and b_2 , are

as follows:

$$\begin{aligned}\theta | b_1 &\sim \text{Gamma}(0.5, b_1) \\ \lambda | b_2 &\sim \text{Gamma}(0.5, b_2) \\ k &\sim \text{Unif}\{1, \dots, 50\} \\ b_1 &\sim \text{Inverse Gamma}(0, 1) \\ b_2 &\sim \text{Inverse Gamma}(0, 1),\end{aligned}$$

where b_1 and b_2 are independent and k , θ , and λ are conditionally independent given b_1 and b_2 . The density functions of the $\text{Gamma}(\alpha, \beta)$ and $\text{Inverse Gamma}(\alpha, \beta)$ distributions are, respectively,

$$f(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-x/\beta} \quad \text{and} \quad f(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{-\alpha-1} e^{-1/(x\beta)}.$$

Please take note: The stated priors for b_1 and b_2 are not proper probability distributions: Integrating $\exp\{-1/b\}/b$ from 0 to ∞ does not converge to a finite value. However, if you simply use the improper prior density $\exp\{-1/b\}/b$, you will obtain proper full conditional distributions for all of the parameters.

Specific instructions are as follows:

- (a) Derive the full conditional densities or mass functions (up to a constant) for each of the five parameters conditional on the other four and the data. For θ , λ , b_1 , and b_2 , describe these full conditionals as coming from some named parametric family, and give the parameters.

Solution: The joint density of the data and the parameters (ignoring multiplicative constants and after some rearranging) is

$$e^{-\theta k} \theta^{S_k} e^{-\lambda(n-k)} \lambda^{S_n-S_k} \frac{1}{(b_1 b_2)^{3/2}} \frac{1}{\sqrt{\theta \lambda}} e^{-1/b_1} e^{-\theta/b_1} e^{-1/b_2} e^{-\lambda/b_2},$$

where $S_k = \sum_{i=1}^k Y_i$ for $1 \leq k \leq n$, and this may be taken as the full conditional density, up to a constant, for each of the five parameters. The dependence of this density on k is somewhat complicated; however, each of the other four parameters has a recognizable full conditional distribution. Isolating each of them in turn from the product above gives

$$\begin{aligned}\text{For } \theta: & \quad \theta^{S_k-1/2} e^{-\theta(k+1/b_1)} & \text{thus, } \theta &\sim \text{Gamma}\left[\frac{1}{2} + S_k, b_1/(1 + b_1 k)\right]. \\ \text{For } \lambda: & \quad \lambda^{S_n-S_k-1/2} e^{-\lambda(n-k+1/b_2)} & \text{thus, } \lambda &\sim \text{Gamma}\left[\frac{1}{2} + S_n - S_k, b_2/(1 + b_2(n - k))\right]. \\ \text{For } b_1: & \quad b_1^{-3/2} e^{-(\theta+1)/b_1} & \text{thus, } b_1 &\sim \text{Inverse Gamma}\left[\frac{1}{2}, 1/(1 + \theta)\right]. \\ \text{For } b_2: & \quad b_2^{-3/2} e^{-(\lambda+1)/b_2} & \text{thus, } b_2 &\sim \text{Inverse Gamma}\left[\frac{1}{2}, 1/(1 + \lambda)\right].\end{aligned}$$

- (b) Implement a one-variable-at-a-time Metropolis-Hastings algorithm to sample from the posterior distribution. For θ , λ , b_1 , and b_2 , use Gibbs sampling from the full conditionals; for k , use a Metropolis-Hastings update where the proposal distribution is uniform on $\{2, \dots, 49\}$. Run at least one million iterations of the algorithm.

Solution: The M-H algorithm is straightforward using the full conditionals in part (a), except the update for k , for which the logarithm of the acceptance probability equals

$$(k - k^*)(\theta - \lambda) + (S_{k^*} - S_k)(\log \theta - \log \lambda)$$

In calculating the ratio above, we use the fact that the uniform proposal distribution is symmetric, so the ratio is simply $\pi(k^*)/\pi(k)$. Here is the code:

```
> m <- 1e6
> # Set up parameter vectors
> Theta <- Lambda <- k <- b1 <- b2 <- rep(1, 1 + m)
```

```

> # Read data (only need second column)
> Y <- read.table("http://sites.stat.psu.edu/~dhunter/515/hw/hw11prob2.txt", head=T)[,2]
> cy <- cumsum(Y) # We only need the cumulative sums in the algorithm
> for (i in 1:m) {
+   # First, the 4 Gibbs sampling steps:
+   Theta[i+1] <- rgamma(1, .5+cy[k[i]], scale=b1[i]/(1+k[i]*b1[i]))
+   Lambda[i+1] <- rgamma(1, .5+cy[50]-cy[k[i]], scale=b2[i]/(1+(50-k[i])*b2[i]))
+   b1[i+1] <- 1/rgamma(1, 1/2, rate=1+Theta[i+1])
+   b2[i+1] <- 1/rgamma(1, 1/2, rate=1+Lambda[i+1])
+   # Now the update of k:
+   kstar <- 1+sample(48,1) # Choose uniformly from 2, ..., 49
+   logratio <- (k[i]-kstar)*(Theta[i+1]-Lambda[i+1]) +
+               (cy[kstar]-cy[k[i]]) * (log(Theta[i+1]) - log(Lambda[i+1]))
+   if (log(runif(1)) < logratio) {
+     k[i+1] <- kstar
+   } else {
+     k[i+1] <- k[i]
+   }
+ }

```

- (c) Give a 95% credible interval for k (use the 0.025 and 0.975 quantiles of the posterior distribution of k). Then, produce a plot of the data (Time vs. Count) and overlay the mean of the Poisson process on the same plot, where this mean is determined by the posterior means of k , θ , and λ .

Solution: Here is the credible interval:

```

> quantile(k, c(.025, .975))
2.5% 97.5%
30    37

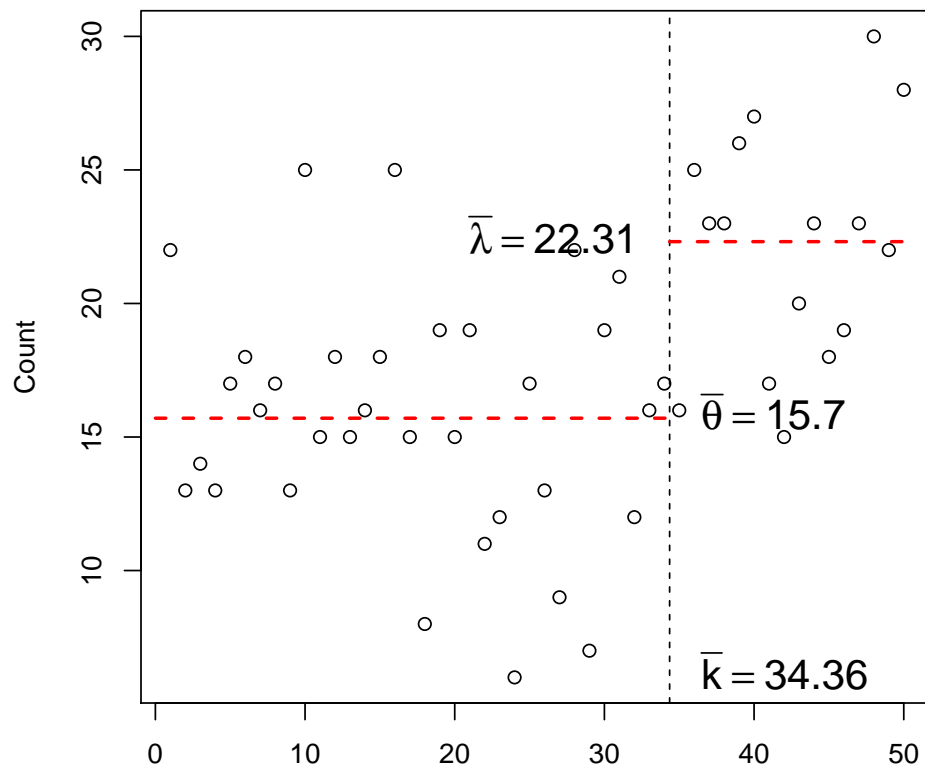
```

In fact, I generated these data using a true value of $k = 35$ (and $\theta = 16$ and $\lambda = 22$), so this procedure based on a sample of only size 50 seems to have worked pretty well. Below is a plot of the data with the appropriate means overlaid. You may find the text-plotting function useful in the future.

```

> plot(Y, ylab="Count", xlab="")
> abline(v=mean(k), lty=2)
> lines(c(0, mean(k)), rep(mean(Theta), 2), lty=2, lwd=2, col=2)
> lines(c(mean(k), 51), rep(mean(Lambda), 2), lty=2, lwd=2, col=2)
> # Add values of posterior means to plot:
> text(mean(k)+1, min(Y),
+       as.expression(bquote(bar(k) == .( round(mean(k),2)  ))),
+       pos=4, cex=1.5)
> text(mean(k)+1, mean(Theta),
+       as.expression(bquote(bar(theta) == .( round(mean(Theta),2)  ))),
+       pos=4, cex=1.5)
> text(mean(k)-1, mean(Lambda),
+       as.expression(bquote(bar(lambda) == .( round(mean(Lambda),2)  ))),
+       pos=2, cex=1.5)

```



If you are interested in seeing an application of the model in this problem to a real dataset, or if you simply get stuck on this problem, you might find the writeup by Murali Haran at <http://sites.stat.psu.edu/~mharan/MCMCtut/MCMC.html> to be helpful.