Gibbs Samplers

Jonathan Navarrete

July 2, 2017

Introduction

Now that we've familiarized ourselves with MCMC and the Metropolis-Hastings algorithms, we begin to analyze a now-common MCMC algorithm called Gibbs sampler. The Gibbs sampler is in fact a special case of the Metropolis-Hastings algorithm for high dimensional target distributions.

We introduce the Gibbs sampler with a two-stage example. The **two-state Gibbs sampler algorithm** as described Robert & Casella goes as follows

`Take` $X_0 = x_0$

For $t = 1, 2, ...,$ `generate`

1. $Y_t \sim f_{Y|X}(\cdot|x_{t-1})$
2. $X_t \sim f_{X|Y}(\cdot|y_t)$

The *two-stage* Gibbs sampler creates a Markov chain from a joint distribution in the following way. If two random variables X and Y have joint density $f(x, y)$, with corresponding conditional densities $f_{Y|X}$ and $f_{X|Y}$, the two stage Gibbs sampler generates a Markov chain (X_t, Y_t) by generating Y_t from conditional density $f_{Y|X}$ and then generating X_t from conditional density $f_{X|Y}$.

We illustrate the implementation of the Gibbs sampler with a simple example. Consider a bivariate Normal distribution where

$$X, Y \sim N_2(\begin{pmatrix}\mu_X \\ \mu_Y\end{pmatrix}, \begin{pmatrix}\sigma_X^2 & \sigma_{XY}^2 \\ \sigma_{YX}^2 & \sigma_Y^2\end{pmatrix})$$

The marginal distributions of X and Y are $N(\mu_X, \sigma_X)$ and $N(\mu_X, \sigma_X)$. The conditional distributions of Y and X are

$$Y|X = x \sim N(\mu_Y + \frac{\rho\sigma_Y}{\sigma_X}(x - \mu_X), (1 - \rho^2)\sigma_Y^2)$$

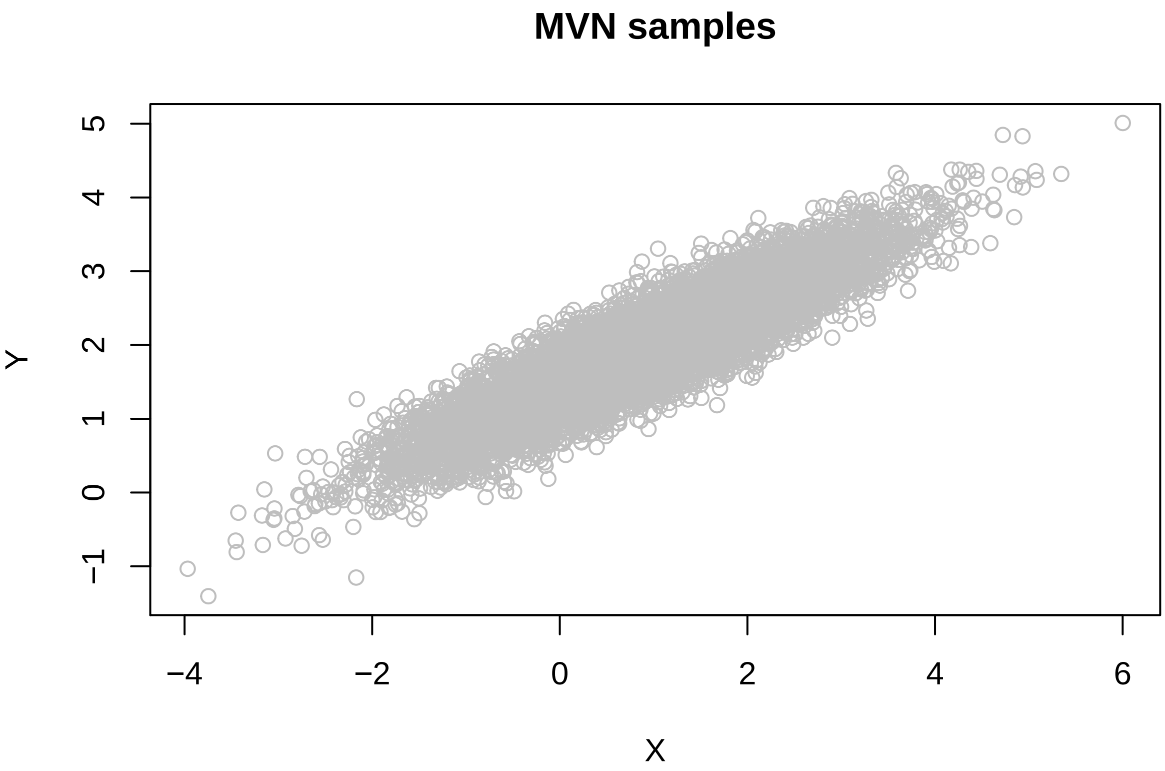
and

$$X|Y = y \sim N(\mu_X + \frac{\rho\sigma_X}{\sigma_Y}(y - \mu_Y), (1 - \rho^2)\sigma_X^2)$$

ρ is the correlation between X and Y, and $(1 - \rho^2)\sigma_X^2$ is the variance.

```
N = 10000

rho = 0.9
mu_x = 1
mu_y = 2
sd_x = 1.2
sd_y = 0.75

s1 = sqrt(1 - rho^2) * sd_x
s2 = sqrt(1 - rho^2) * sd_y

MVN = matrix(data = NA, nrow = N, ncol = 2,
          dimnames = list(NULL, c("X", "Y")))

MVN[1, ] = c(mu_x, mu_y)
```

```
Y = MVN[1, 2] ## get Y vals
for(i in 1:(N-1)){
  mx = mu_x + rho * (Y - mu_y) * sd_x/sd_y
  X = rnorm(n = 1, mx, s1)
  MVN[i+1, 1] = X
  my = mu_y + rho * (X - mu_x) * sd_y/sd_x
  Y = rnorm(n = 1, mean = my, sd = s2)
  MVN[i+1, 2] = Y
}


plot(MVN, type = "p", col = 8,
     main = "MVN samples")
```



```
## means
colMeans(MVN)
```

```
##         X         Y
## 0.9178736 1.9474894
```

```
## correlation
cor(MVN)
```

```
##           X         Y
## X 1.0000000 0.9025664
## Y 0.9025664 1.0000000
```

Beta-Binomial revisited

In the introduction to these notes, we saw a Bayesian example of the Beta-Binomial distribution. From Casella's paper *Explaining the Gibbs Sampler*, we revisit this example.

$$X|\theta \sim Bin(n, \theta), \text{ and } \theta \sim Beta(a, b)$$

have joint density

$$f(x, \theta) = \binom{n}{x} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{x+a-1}(1-\theta)^{n-x+b-1}$$

is the a $Beta(x + a, n - x + b)$ distribution.

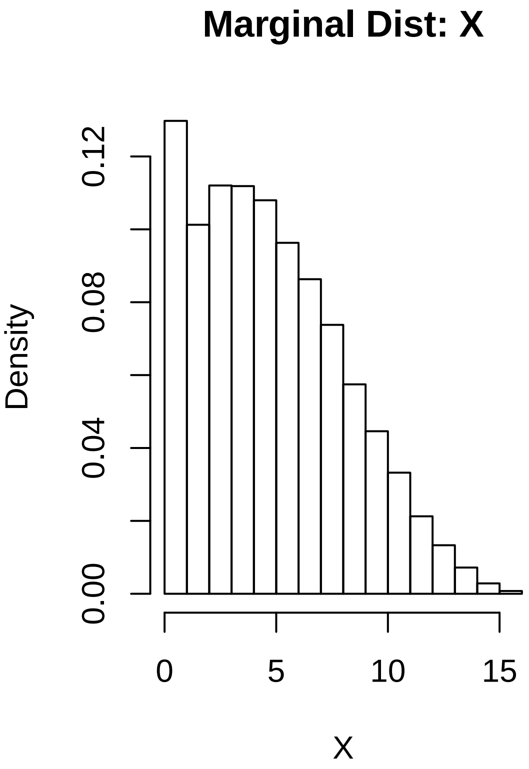
Suppose we are interested in calculating some characteristics of the marginal distributions of $X|\theta$ and $\theta|a, b, x, n$.

$$f(x|\theta) \text{ is } Bin(n, \theta) f(\theta|x) \text{ is } Beta(x + a, n - x + b)$$

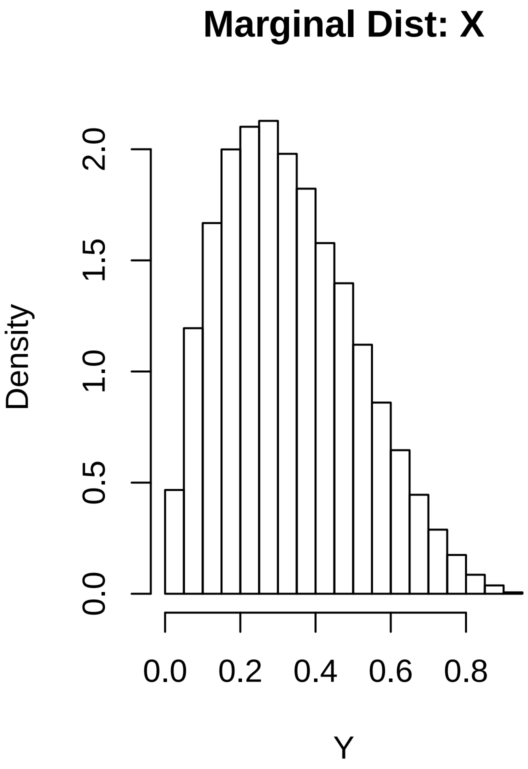
Therefore, we follow an iterative algorithm of

$$X_i \sim f(x|\theta) Y_{i+1} \sim f(y|X_i = x_i)$$

```r
N = 10^5
n = 16
a = 2
b = 4
X = numeric(N)
Y = numeric(N)

## initial values
X[1] = 0.2
Y[1] = 0.34 ## theta values

for(i in 1:N){
  X[i+1] = rbinom(n = 1, size = n, prob = Y[i])
  Y[i+1] = rbeta(1, a + X[i+1], n - X[i+1]+b)
}

par(mfrow = c(1,2))
hist(X, main = "Marginal Dist: X", probability = TRUE)
hist(Y, main = "Marginal Dist: X", probability = TRUE)
```



Marginal Dist: X
Density
0.00
0.04
0.08
0.12
0
5
10
15
X



Marginal Dist: X
Density
0.0
0.5
1.0
1.5
2.0
0.0
0.2
0.4
0.6
0.8
Y

Poisson-Gamma Model

Consider a random sample $\mathbf{y} = (y_1, ..., y_n)^T$ where $y_i|\theta \sim Poisson(\theta)$ where $\theta \sim Gamma(\alpha, \beta)$ for $i \in 1, ..., k$. Assume t and α are known constants, but that β has a Gamma, $Gamma(c, d)$, hyperprior with known hyperparameters c and d. This represents a three stage model where the likelihood, prior and hyperprior are defined as

$$f(y_i|\theta) = \frac{\theta^{y_i}}{y_i!}e^{-\theta}, y_i \geq 0, \theta_i > 0$$

$$g(\theta|\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}, \alpha > 0, \beta > 0$$

$$h(\beta) = \frac{d^c}{\Gamma(c)}\beta^{c-1}e^{-d\beta}, c > 0, d > 0$$

Suppose our interest is in generating samples from the marginal posterior distributions of θ, $p(\theta|\mathbf{y})$. Note that while the gamma prior is conjugate with the Poisson likelihood and the gamma hyperprior is conjugate with the gamma prior, no closed for for $p(\theta|\mathbf{y})$ exists. However, the full conditional distributions of β and the θ needed to implement the Gibbs sampler can be extracted from the full conditional $p(\theta|\mathbf{y})$

$$\begin{aligned}
p(\theta|\mathbf{y}, \beta) &\propto \prod_{i=1}^{n} f(y_i|\theta)g(\theta|\beta) \\
&\propto \theta^{n\bar{y}}e^{n\theta} \times \theta^{\alpha-1}e^{-\theta\beta} \\
&\propto \theta^{n\bar{y}+\alpha-1}e^{-\theta(n+\beta)} \\
&\propto Gamma(\theta|n\bar{y} + \alpha, n + \beta)
\end{aligned}$$

while for β we have

$$\begin{aligned}
p(\beta|\theta, \mathbf{y}) &\propto g(\theta|\beta) \times h(\beta) \\
&= \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta} \times \frac{d^c}{\Gamma(c)}\beta^{c-1}e^{-d\beta} \\
&\propto \beta^{\alpha+c-1}e^{-\beta(\theta+d)} \\
&\propto Gamma(\alpha + c, \theta + d)
\end{aligned}$$

Bayesian Analysis with Normal Data

Suppose we have a random sample $y_1, ..., y_n|\mu, \tau \sim N(\mu, 1/\tau)$. Here we use a precision parameter τ instead of σ for standard devation (or σ^2 for variance) . Place two independent priors on the unkown parameters μ and τ.

$$\mu \sim N(a, 1/b) \perp \tau \sim Gamma(c, d)$$

The posterior is given by

$$p(\mu, \tau|\mathbf{y}) = \frac{L(\mu, \tau|\mathbf{y}) \times p(\mu, \tau)}{m(\mathbf{y})}$$

where $p(\mu, \tau) = p(\mu) \times p(\tau)$ and $m(\mathbf{y}) = \int L(\mu, \tau|\mathbf{y}) \times p(\mu, \tau)$

The likelihood is

$$\begin{aligned} L(\mu, \tau|\mathbf{y}) &= \prod_{i=1}^{n} f(y_i|\mu, \tau) \\ &= \frac{\tau}{2\pi}^{n/2} exp(\frac{\tau}{2} \sum (y_i - \mu)^2) \\ &= \frac{\tau}{2\pi}^{n/2} exp\frac{\tau}{2}(n-1)s^2 - \frac{\tau}{2}(\bar{y} - \mu)^2 \text{ (obtained after some algebra)} \\ &\propto \tau^{n/2} exp\frac{\tau}{2}(n-1)s^2 - \frac{\tau}{2}(\bar{y} - \mu)^2 \text{ (drop unimportant constants)} \end{aligned}$$

The posterior is then given by

$$\begin{aligned} p(\mu, \tau) &\propto L(\mu, \tau|\mathbf{y}) \times p(\mu, \tau) \\ &= \tau^{n/2} exp(\frac{\tau}{2}(n-1)s^2 - \frac{\tau}{2}(\bar{y} - \mu)^2) \times \frac{d^c}{\Gamma(c)} \tau^{c-1} exp(-b\tau) \times \sqrt{\frac{b}{2\pi}} exp(-\frac{b}{2}(\mu - a)^2) \\ &\text{(next, drop unimportant constants)} \\ &\propto \tau^{n/2} exp(\frac{\tau}{2}(n-1)s^2 - \frac{\tau}{2}(\bar{y} - \mu)^2) \times \tau^{c-1} exp(-b\tau) \times \ exp(-\frac{b}{2}(\mu - a)^2) \\ &= \tau^{n/2+c-1} exp(\frac{\tau}{2}(n-1)s^2 - \frac{\tau}{2}(\bar{y} - \mu)^2) \times exp(-b\tau) \times \ exp(-\frac{b}{2}(\mu - a)^2) \end{aligned}$$

And the final result is some function that is not recognizable as any commonly known distribution. Hence, we're stuck if we are to try to sample from this distribution directly!

However, if we are too look at the conditional posteriors, $\mu|\tau, \mathbf{y}$

$$p(\mu|\tau, \mathbf{y}) \propto exp(-\frac{1}{2}(n\tau + b)(\mu - \hat{\mu}_\tau)^2)$$

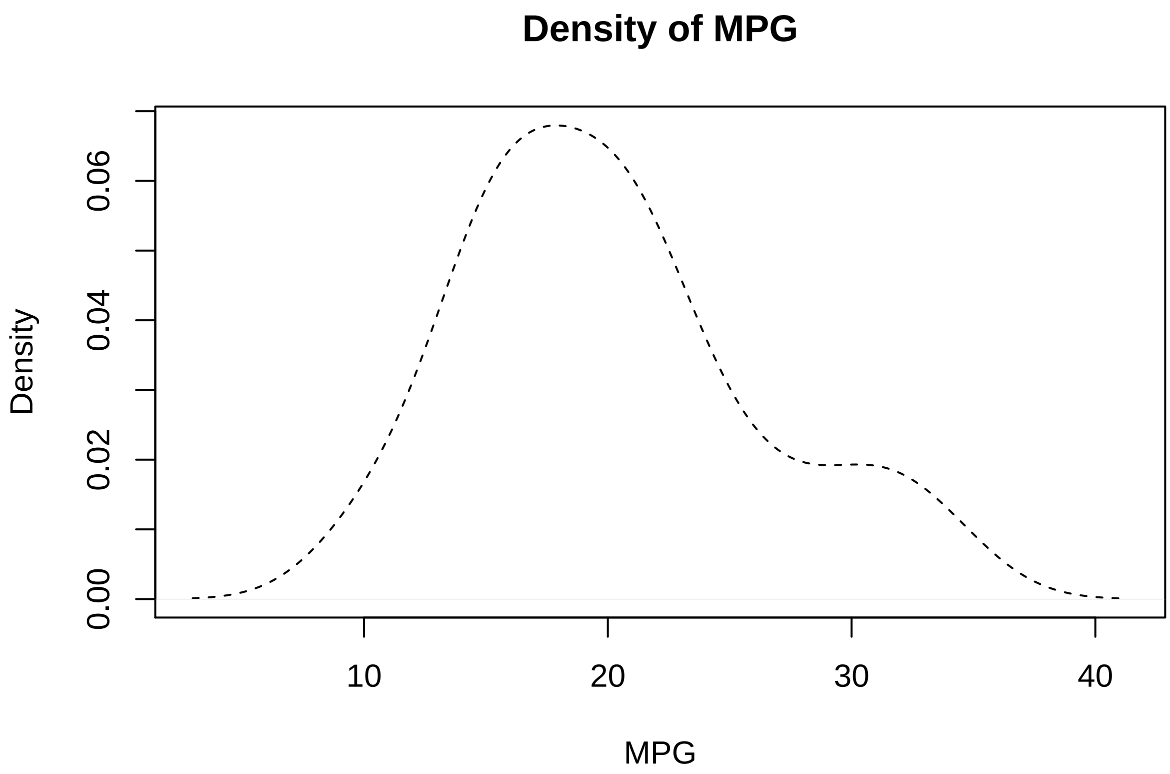
and $\tau|\mu, \mathbf{y}$

$$p(\tau|\mu, \mathbf{y}) \propto \tau^{(n/2+c)-1} exp(-\tau(d + \frac{n(\bar{y} - \mu)^2 + (n-1)s^2}{2}))$$

we find that there are recognizable distributions. They are $N(\hat{\mu}, \frac{1}{n\tau+b})$ and $Gamma(\frac{n}{2} + c, d + \frac{n(\bar{y}-\mu)^2+(n-1)s^2}{2})$

Given that we can sample from the conditional distributions, we can utilize a Gibbs sampler to sample from the posterior distribution $p(\mu, \tau|\mathbf{y})$.

```r
y = mtcars$mpg

plot(density(y), main = "Density of MPG",
     lty = 2, xlab = "MPG")
```



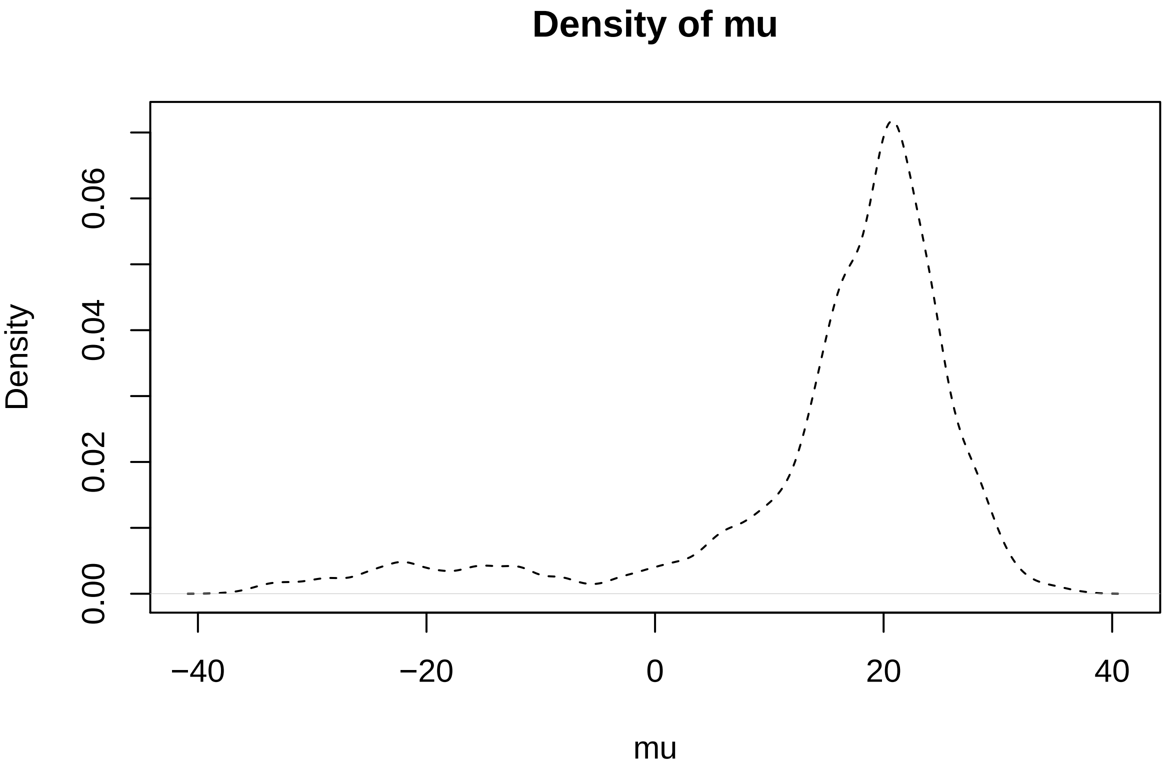
```
# This function takes the prior mean and a percentile and returns the corresponding hyperparameter valu
find.normal <- function(prior.mean, percentile, p=0.95){
  prior.sd <- (qnorm(p) / (percentile - prior.mean))^{-1}
  return(list(mean=prior.mean, sd=prior.sd, precision=1/prior.sd^2))
}


# This function takes the prior mode and a percentile of sigma and returns the corresponding parameters
find.tau.gamma <- function(prior.sigma.mode, sigma.percentile, p=0.95, start.a=1, end.a=1000, increment
  a <- seq(from=start.a, to=end.a,  by=increment)
  b <- prior.sigma.mode^2 * (a + 1)
  i <- which.max(qgamma(1-p, a, b) > 1/sigma.percentile^2)
  return(list(a=a[i], b=b[i]))
}
```

```
N = 3 * 10^3
n = length(y)
ybar = mean(y)
s = sd(y)
mu = numeric(N)
tau = numeric(N)

## initial values
mu = ybar
tau = 1/sd(y)


## Normal hyper parameters
a <- find.normal(prior.mean=21, percentile = 24, p=0.95)$mean
b <- find.normal(prior.mean=21, percentile = 24, p=0.95)$precision
```

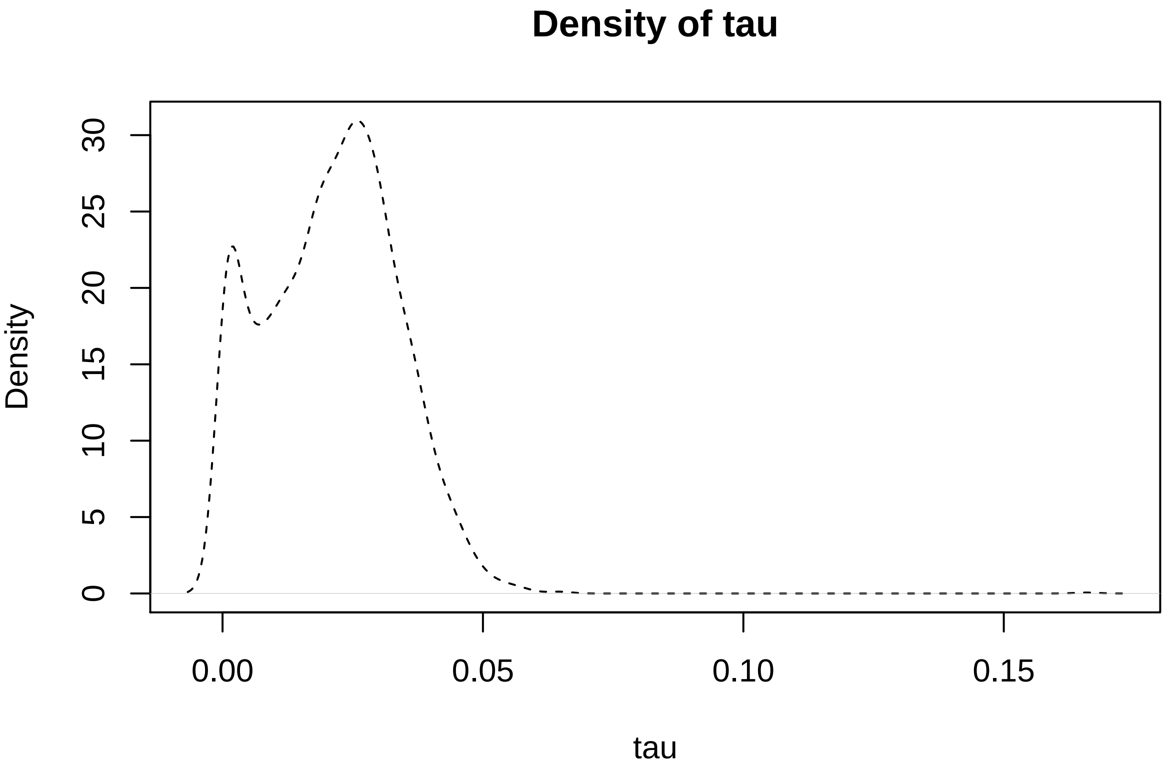
```
## Gamma Hyper parameters
gamma_HyperParams <- find.tau.gamma(prior.sigma.mode= 1.2, sigma.percentile = 2.462, p=0.95) # Returns
gamma_HyperParams
```

```
## $a
## [1] 3.46
##
## $b
## [1] 6.4224
```

```
c = gamma_HyperParams$a ## arbitrary
d = gamma_HyperParams$b ## arbitrary

for(i in 2:N){
  SHAPE = c + n/2
  RATE = d + 0.5*( n*(ybar - mu[i-1])^2 + (n-1)*s^2  )
  tau[i]  = rgamma(n = 1, shape = SHAPE, rate = RATE)

  MU = mu[i-1]
  SD = 1/(n*tau[i] + b)
  mu[i] = rnorm(n = 1, mean = MU, sd = SD)
}

B = 10^2

muPosterior = mu[-(1:B)]
tauPosterior = tau[-(1:B)]

plot(density(mu), main = "Density of mu",
     lty = 2, xlab = "mu")
```



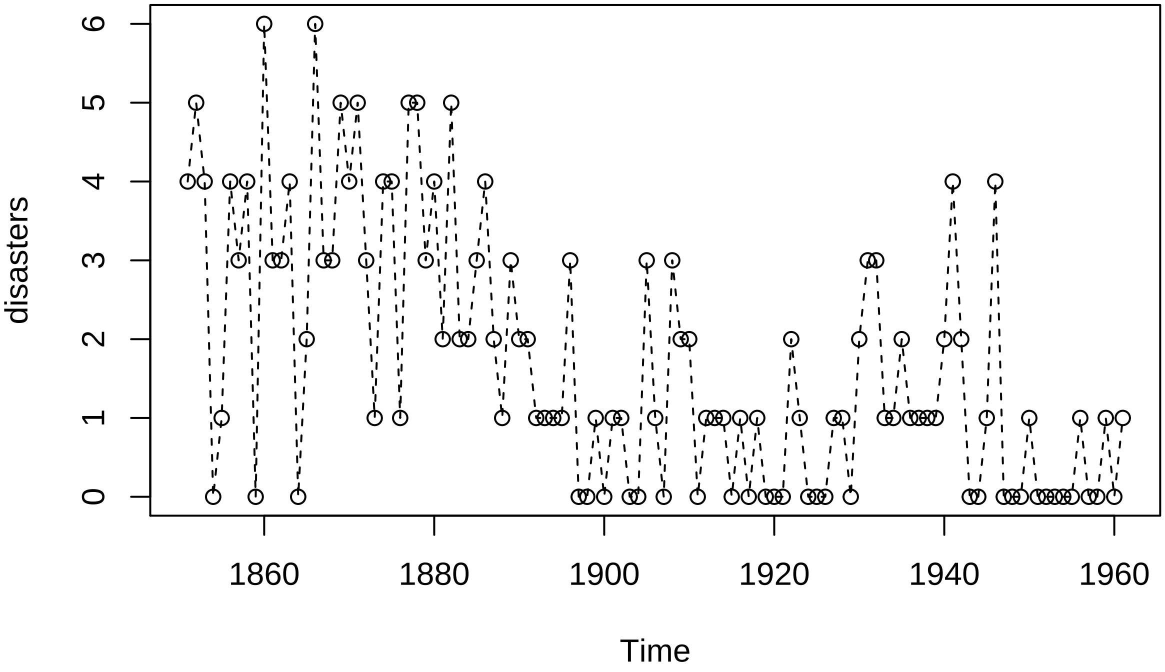
```r
plot(density(tau), main = "Density of tau",
     lty = 2, xlab = "tau")
```



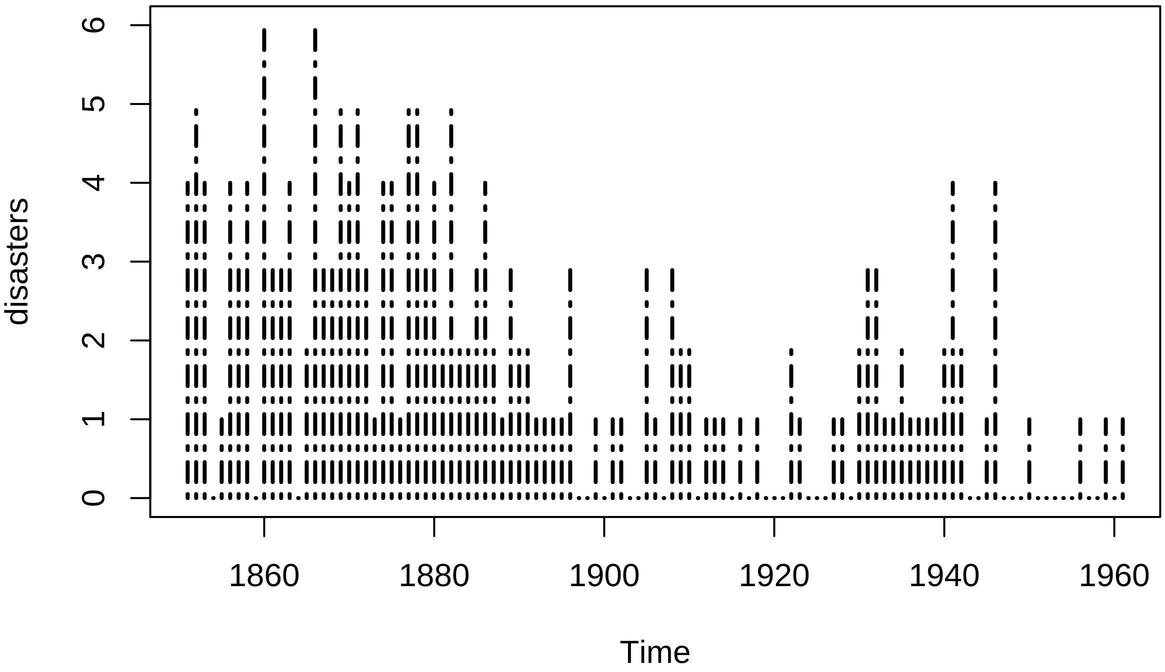
Example: Inferring patterns in UK coal mining disasters

Let’s try to model a more interesting example, a time series of recorded coal mining disasters in the UK from 1851 to 1962. Occurrences of disasters in the time series is thought to be derived from a Poisson process with a large rate parameter in the early part of the time series, and from one with a smaller rate in the later part. We are interested in locating the change point in the series, which perhaps is related to changes in mining safety regulations.

```
data_array = c(4, 5, 4, 0, 1, 4, 3, 4, 0, 6, 3, 3, 4, 0, 2, 6,
                        3, 3, 5, 4, 5, 3, 1, 4, 4, 1, 5, 5, 3, 4, 2, 5,
                        2, 2, 3, 4, 2, 1, 3, 2, 2, 1, 1, 1, 1, 3, 0, 0,
                        1, 0, 1, 1, 0, 0, 3, 1, 0, 3, 2, 2, 0, 1, 1, 1,
                        0, 1, 0, 1, 0, 0, 0, 2, 1, 0, 0, 0, 1, 1, 0, 2,
                        3, 3, 1, 1, 2, 1, 1, 1, 1, 2, 4, 2, 0, 0, 1, 4,
                        0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1)

disasters = ts(data_array, freq=1, start = 1851)

plot(disasters, type = "o", lty = 2)
```



```
plot(disasters, type = "h", lty = 6, lwd = 2)
```



We are going to use Poisson random variables for this type of count data. Denoting year i's accident count by y_i,

$$y_i \sim \text{Poisson}(\lambda)$$

The modeling problem revolves around estimating the values of the λ parameters. Looking at the time series above, it appears that the rate declines later in the time series. A changepoint model identifies a point (year) during the observation period (call it τ) after which the parameter λ drops to a lower value. So we are estimating two λ parameters: one for the early period and another for the late period.

$$\lambda = \begin{cases} \lambda_1 \text{ if } t < \tau \\ \lambda_2 \text{ if } t \geq \tau \end{cases}$$

We need to assign prior probabilities to both λ parameters. The gamma distribution not only provides a continuous density function for positive numbers, but it is also conjugate with the Poisson sampling distribution.

We will specify suitably vague hyperparameters α and β for both priors.

$$\lambda_1 \sim \text{Gamma}(1, \beta_1)$$
$$\lambda_2 \sim \text{Gamma}(1, \beta_2)$$

Since we do not have any intuition about the location of the changepoint (prior to viewing the data), we will assign a discrete uniform prior over all years 1851-1962.

$$\tau \sim \text{DiscreteUniform(1851,1962)}$$

$$\Rightarrow P(\tau = k) = \frac{1}{111}$$

Implementing Gibbs sampling We are interested in estimating the joint posterior of λ_1, λ_2 and τ given the array of annnual disaster counts $\mathbf{y}$. This gives:

$$P(\lambda_1, \lambda_2, \tau | \mathbf{y}) \propto P(\mathbf{y} | \lambda_1, \lambda_2, \tau) P(\lambda_1, \lambda_2, \tau)$$

To employ Gibbs sampling, we need to factor the joint posterior into the product of conditional expressions:

$$P(\lambda_1, \lambda_2, \tau | \mathbf{y}) \propto P(y_{t<\tau} | \lambda_1, \tau) P(y_{t \geq \tau} | \lambda_2, \tau) P(\lambda_1) P(\lambda_2) P(\tau)$$

which we have specified as:

$$\begin{aligned}
P(\lambda_1, \lambda_2, \tau | \mathbf{y}) &\propto \left[\prod_{t=1851}^{\tau} \text{Poi}(y_t | \lambda_1) \prod_{t=\tau+1}^{1962} \text{Poi}(y_t | \lambda_2) \right] \text{Gamma}(\lambda_1 | \alpha, \beta) \text{Gamma}(\lambda_2 | \alpha, \beta) \frac{1}{111} \\
&\propto \left[\prod_{t=1851}^{\tau} e^{-\lambda_1} \lambda_1^{y_t} \prod_{t=\tau+1}^{1962} e^{-\lambda_2} \lambda_2^{y_t} \right] \lambda_1^{\alpha-1} e^{-\beta\lambda_1} \lambda_2^{\alpha-1} e^{-\beta\lambda_2} \\
&\propto \lambda_1^{\sum_{t=1851}^{\tau} y_t + \alpha - 1} e^{-(\beta+\tau)\lambda_1} \lambda_2^{\sum_{t=\tau+1}^{1962} y_i + \alpha - 1} e^{-\beta\lambda_2}
\end{aligned}$$

So, the full conditionals are known, and critically for Gibbs, can easily be sampled from.

$$\lambda_1 \sim \text{Gamma}(\sum_{t=1851}^{\tau} y_t + \alpha, \tau + \beta)$$

$$\lambda_2 \sim \text{Gamma}(\sum_{t=\tau+1}^{1962} y_i + \alpha, 1962 - \tau + \beta)$$

$$\tau \sim \text{Categorical}\left(\frac{\lambda_1^{\sum_{t=1851}^{\tau} y_t + \alpha - 1} e^{-(\beta+\tau)\lambda_1} \lambda_2^{\sum_{t=\tau+1}^{1962} y_i + \alpha - 1} e^{-\beta\lambda_2}}{\sum_{k=1851}^{1962} \lambda_1^{\sum_{t=1851}^{\tau} y_t + \alpha - 1} e^{-(\beta+\tau)\lambda_1} \lambda_2^{\sum_{t=\tau+1}^{1962} y_i + \alpha - 1} e^{-\beta\lambda_2}} \right)$$

Implementing this in Python requires random number generators for both the gamma and discrete uniform distributions. We can leverage NumPy for this:

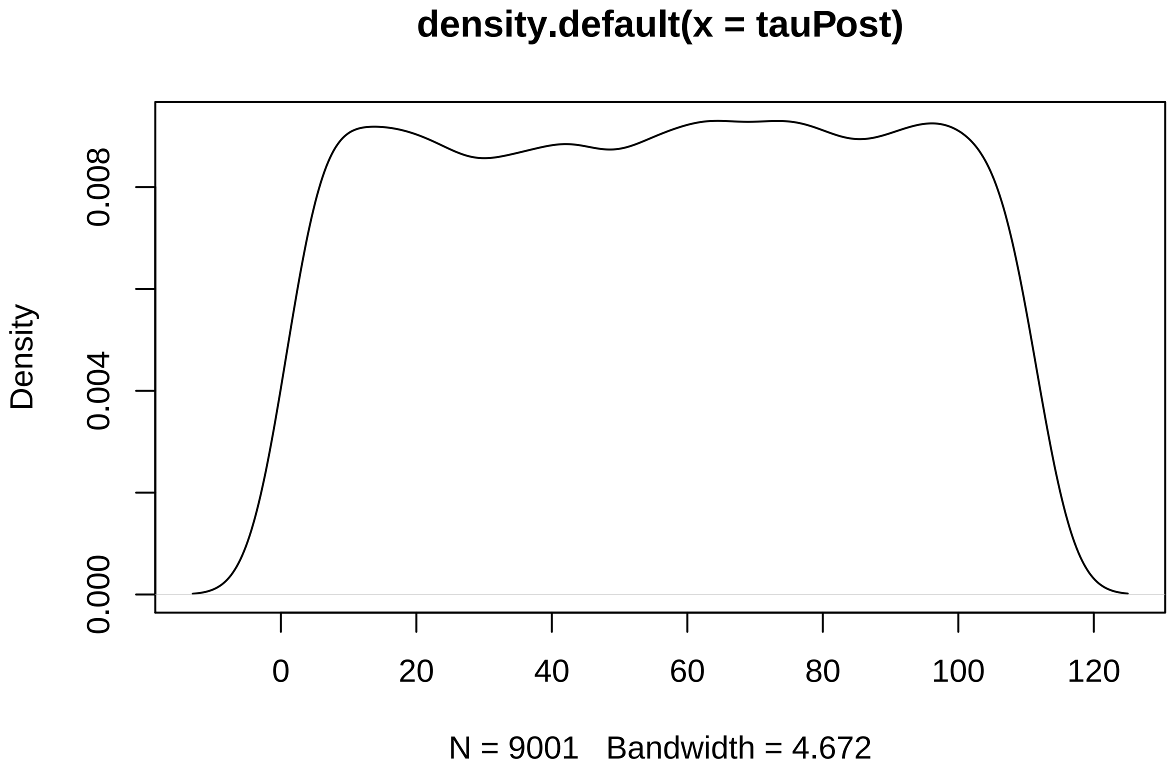
Function to draw random gamma variate

```
alpha = 1.0
beta = 10

# Specify number of iterations
N = 10^4
B = 1000

# Initialize trace of samples
lambda1 = numeric(N+1)
lambda2 = numeric(N+1)
tau = numeric(N+1)


## initialize values
lambda1[1] = 6
lambda2[1] = 2
tau[1] = 50


n_count_data = length(disasters)
```

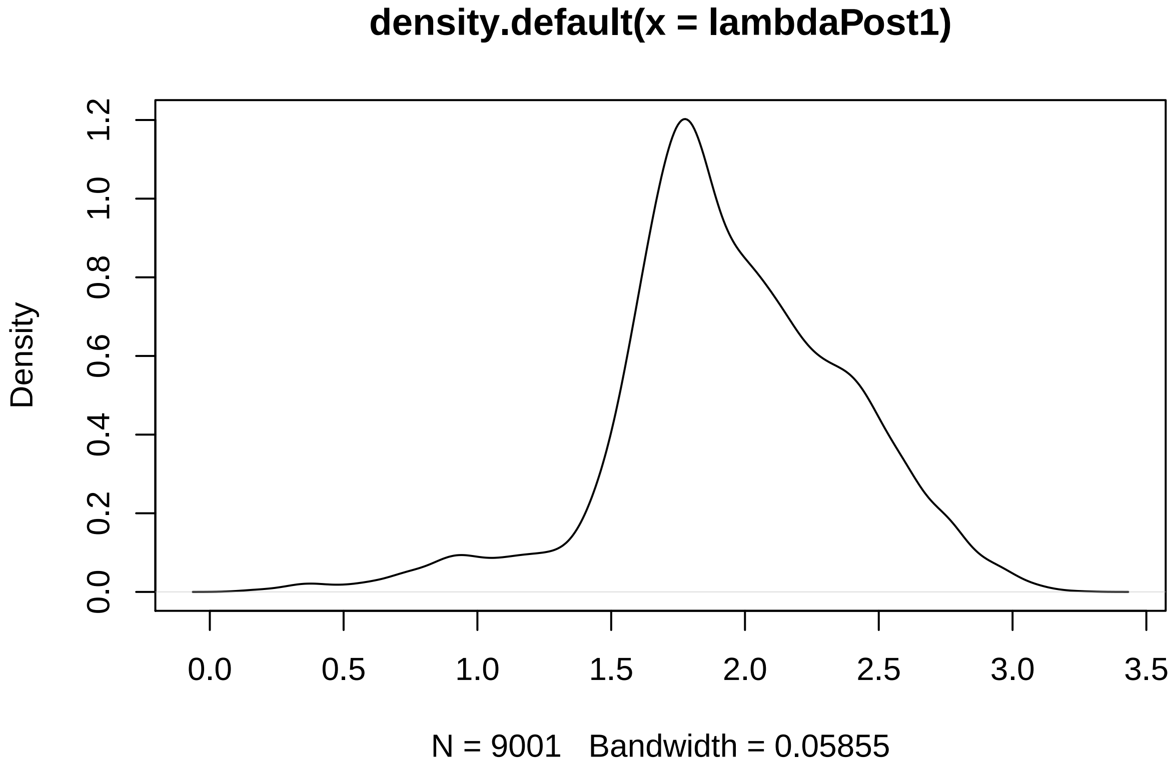
```
DGamma = function(lambda, a, b){
  lambda**(a-1) * exp(-b*lambda)
}
```

```
disasters_array = disasters
# Sample from conditionals
for(i in 1:N){
  # Sample early mean
  sumDisasters = sum(disasters_array[1:tau[i]])  ## disasters_array[tau[i]:].sum()
  lambda1[i+1] = rgamma(n = 1, shape = sumDisasters + alpha, scale = 1/(tau[i] + beta))

  # Sample late mean
  lambda2[i+1] = rgamma(n = 1, shape = sumDisasters + alpha, scale = 1/ (n_count_data - tau[i] + beta))

  # Sample changepoint: first calculate probabilities (conditional)
  p = numeric(n_count_data)
  for(t in 1:n_count_data){
   G1 = DGamma(lambda1[i+1], sum(disasters_array[1:t]) + alpha, t + beta)
   G2 = DGamma(lambda2[i+1], sum(disasters_array[(t+1):111]) + alpha, n_count_data - t + beta)
   p[t] = s1*s2
  }
  # # ... then draw sample
 tau[i+1] = sample(x = 1:111, size = 1, prob = p/sum(p))
  #runif(n = 1, min = 1, max = 111)  #rcategorical(p/p.sum())
}
```

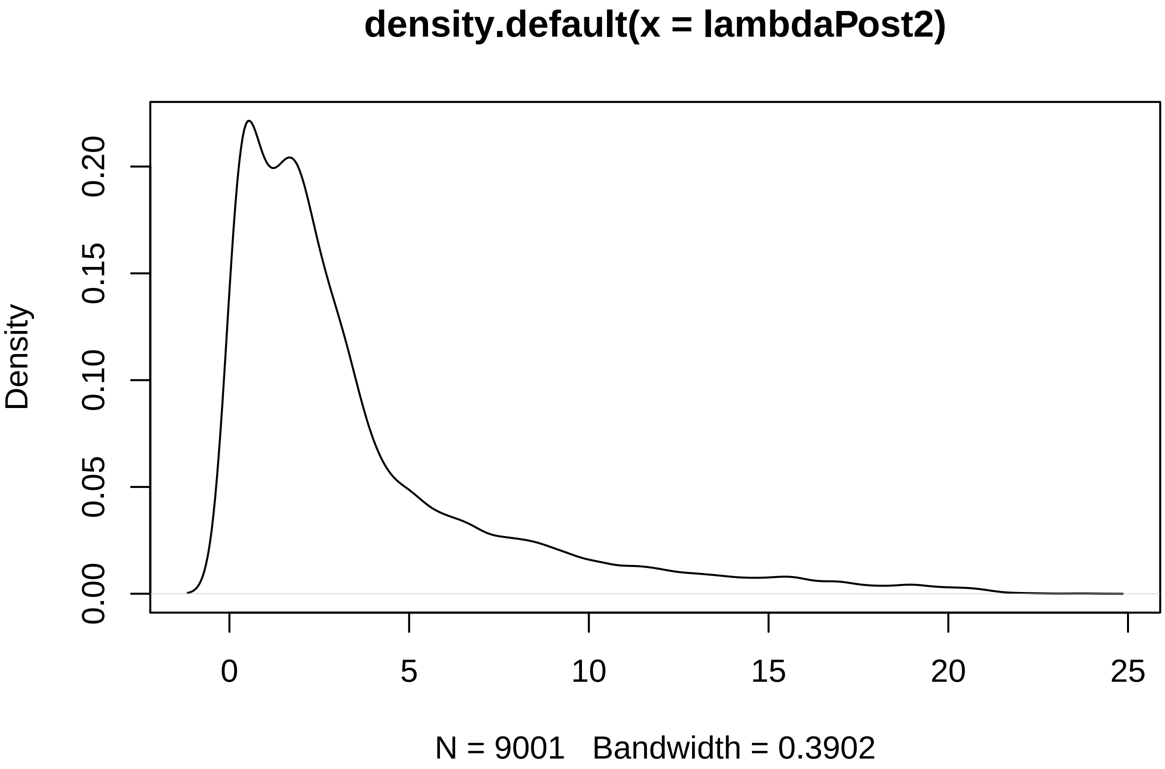
```
tauPost = tau[-(1:B)]
lambdaPost1 = lambda1[-(1:B)]
lambdaPost2 = lambda2[-(1:B)]

plot(density(tauPost))
```



```r
plot(density(lambdaPost1))
```



```r
plot(density(lambdaPost2))
```


density.default(x = lambdaPost2)
Density
0.00
0.05
0.10
0.15
0.20
0
5
10
15
20
25
N = 9001   Bandwidth = 0.3902