Exercise 1: Let the discrete random variable X be described by a probability mass function $p_X(x) = \Pr(X = x)$. The current state of a Metropolis–Hastings Markov chain is x_t, which is generated from the same distribution as X. Demonstrate that the next state x_{t+1} will also be drawn from the same distribution as X.

Solution: Let's assume the following labels for the states of the Markov chain: $x_{t+1} = x_b$ and $x_t = x_a$. The Metropolis–Hastings ratio is

$$r(x_b \mid x_a) = \frac{q(x_a \mid x_b)p(x_b)}{q(x_b \mid x_a)p(x_a)} \,.$$

The joint probability that $X_{t+1} = x_b$ and $X_t = x_a$ can be decomposed as

$$\Pr(X_{t+1} = x_b, X_t = x_a) = \Pr(X_{t+1} = x_b \mid X_t = x_a)\Pr(X_t = x_a) \,.$$

The Markov chain reaches state x_b at time $t+1$ if this state is proposed and accepted. Let's assume that $r(x_b \mid x_a) > 1$ and consequently $r(x_a \mid x_b) < 1$ so that

$$\Pr(X_{t+1} = x_b, X_t = x_a) = \min[1, r(x_b \mid x_a)]q(x_b \mid x_a)\Pr(X_t = x_a) = q(x_b \mid x_a)p(x_a)$$

On the other hand,

$$\begin{aligned}
\Pr(X_{t+1} = x_a, X_t = x_b) &= \Pr(X_{t+1} = x_a \mid X_t = x_b)\Pr(X_t = x_b) \\
&= \min[1, r(x_a \mid x_b)]q(x_a \mid x_b)p(x_b) \\
&= \frac{q(x_b \mid x_a)p(x_a)}{q(x_a \mid x_b)p(x_b)} q(x_a \mid x_b)p(x_b) \\
&= q(x_b \mid x_a)p(x_a) \\
&= \Pr(X_{t+1} = x_b, X_t = x_a)
\end{aligned}$$

implying that $\Pr(X_{t+1} = x_b, X_t = x_a) = \Pr(X_{t+1} = x_a, X_t = x_b)$. Marginalization of the joint distribution shows that $\Pr(X_{t+1} = x_b) = \Pr(X_t = x_b)$ and since X_t follows the same distribution as X, the next random variable X_{t+1} follows that distribution as well.

Exercise 2 (chapter 7.4): Let the random variable X follow a Laplace distribution with location $\mu = 0$ and scale parameter $\sigma = 2$. The density of the Laplace distribution is

$$f_X(x) = \frac{1}{2\sigma} \exp\left\{-\frac{|x - \mu|}{\sigma}\right\} \qquad \sigma > 0 \,.$$

1. Implement an independent Metropolis–Hastings sampler with a $\text{Normal}\big(0, \sigma_1^2\big)$ proposal distribution.
2. Implement a random walk Metropolis–Hastings sampler based on $\text{Normal}\big(0, \sigma_2^2\big)$ noise.

3. Compare the performance of both samplers in terms of $\mathbb{E}[X]$ and $\mathbb{V}[X]$ for various values of σ_1^2 and σ_2^2. What value of σ_2^2 is required to achieve an acceptance rate of about 40% in case of the random walk Metropolis–Hastings sampler?

Solution: An implementation of the independent Metropolis–Hastings sampler in R is:

```
target <- function( x ) { -0.5 * abs( x ) }
proposal <- function( x, scale ) {  dnorm( x, 0, scale, TRUE ) }
nSamples <- 10000 ; nAccepted <- 0
x <- numeric( nSamples ) ; sigma1 <- 6;
for( ii in seq( 2, nSamples ) ) {
   x[ ii ] <- rnorm( 1, 0, sigma1 )
   alpha <- exp(
      proposal( x[ ii - 1 ], sigma1 ) + target( x[ ii ] ) -
      proposal( x[ ii ], sigma1 ) - target( x[ ii - 1 ] )
   )
   if( runif( 1 ) > alpha ) {
      x[ ii ] <- x[ ii - 1 ]
   } else {
      nAccepted <- nAccepted + 1
   }
}
```

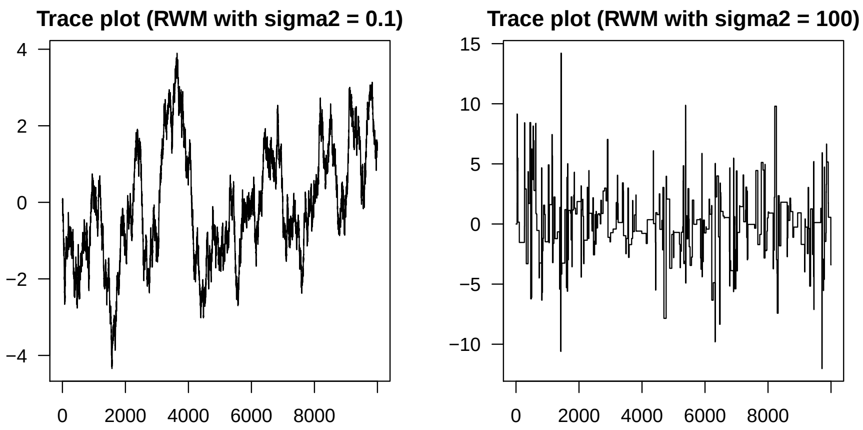
An implementation of the random walk Metropolis–Hastings sampler in R is:

```
target <- function( x ) { -0.5 * abs( x ) }
nSamples <- 10000 ; nAccepted <- 0
x <- numeric( nSamples ) ; sigma2 <- 6
for( ii in seq( 2, nSamples ) ) {
   x[ ii ] <- rnorm( 1, x[ ii - 1 ], sigma2 )
   alpha <- exp( target( x[ ii ] ) - target( x[ ii - 1 ] ) )
   if( runif( 1 ) > alpha ) {
      x[ ii ] <- x[ ii - 1 ]
   } else {
      nAccepted <- nAccepted + 1
   }
}
```

A comparison of both samplers for different values of σ_1^2 and σ_2^2 is shown below. In case of the random walk Metropolis–Hastings sampler, the acceptance rate is high for small values of σ_2^2 (first trace plot). Successive states of the Markov chain are very similar which results in a slow exploration of the distribution and convergence to it. If σ_2^2 is too large (second trace plot), then the proposed states are likely in regions with low probability density which also results in a slow exploration of the distribution and convergence to it.

The value of σ_2^2 has to be between 25 and 100 to achieve an acceptance rate of about 40%. Since $\mathbb{V}[X] = 8$, $\sigma_2^2 = 2.38^2 \cdot 8 \approx 45$ (chapter 7.4.3) results in an acceptance rate of about 40%.

```
#       E[X] (RW) Var[X] (RW) Accepted (RW) E[X] (IND) Var[X] (IND) Accepted (IND)
# 0.01  -0.081406       2.287        0.9798    0.03763      0.04949         0.4255
# 25    -0.059631       7.860        0.4610   -0.02733      7.62242         0.5550
# 100   -0.101546       7.929        0.2756   -0.05885      7.75144         0.3074
# 2500  -0.008731       8.329        0.0677   -0.03320      7.49225         0.0603
# 10000 -0.050212       6.848        0.0295   -0.23956      7.13693         0.0311
```



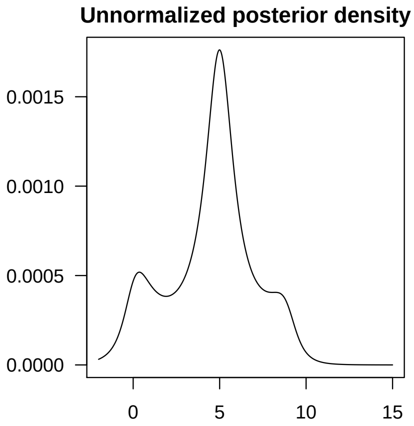
Exercise 3 (chapter 7.4): Let $\{Y_i\}_{i=1}^{3}$ be independent and identically distributed random variables that follow a Cauchy distribution with location μ and scale parameter $\sigma = 1$. The density of the Cauchy distribution is

$$f_Y(y) = \frac{1}{\pi}\left[\frac{\sigma}{\sigma^2 + (y-\mu)^2}\right] \qquad \sigma > 0\,.$$

The prior density of the location parameter is $p(\mu) \propto \exp\{-\mu^2/100\}$.

1. Show that the posterior density has three modes when $Y_1 = 0, Y_2 = 5$ and $Y_3 = 9$.
2. Implement a random walk Metropolis–Hastings sampler based on Cauchy$(0, \sigma_1^2)$ and Normal$(0, \sigma_2^2)$ noise.
3. Compare the performance of both samplers in terms of $\mathbb{E}[\mu \mid y_1, y_2, y_2]$ and monitor convergence using cumulative average plots.

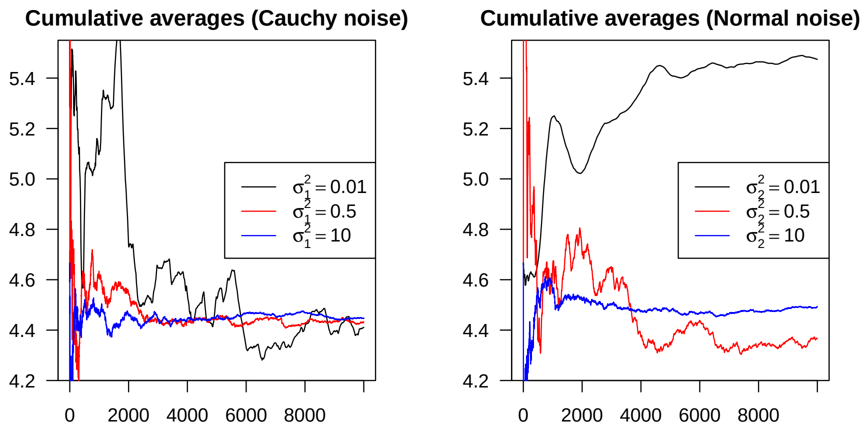
Solution: Trimodality could be checked visually.



An implementation of both random walk Metropolis–Hastings samplers in R is:

```
target <- function( mu, y ) {
   -mu ^ 2 / 100 -
   log( 1 + ( y[ 1 ] - mu ) ^ 2 ) -
   log( 1 + ( y[ 2 ] - mu ) ^ 2 ) -
   log( 1 + ( y[ 3 ] - mu ) ^ 2 )
}
nSamples <- 10000 ;
x <- matrix( 0, nSamples, 2 ) ; y <- c( 0, 5, 9 ) ; sigma <- 0.01
for( ii in seq( 2, nSamples ) ) {
   x[ ii, 1 ] <- rnorm( 1, x[ ii - 1, 1 ], sigma )
   x[ ii, 2 ] <- rcauchy( 1, x[ ii - 1, 2 ], sigma )
   alpha <- c(
      exp( target( x[ ii, 1 ], y ) - target( x[ ii - 1, 1 ], y ) ),
      exp( target( x[ ii, 2 ], y ) - target( x[ ii - 1, 2 ], y ) )
   )
   if( runif( 1 ) > alpha[ 1 ] ) {
      x[ ii, 1 ] <- x[ ii - 1, 1 ]
   }
   if( runif( 1 ) > alpha[ 2 ] ) {
      x[ ii, 2 ] <- x[ ii - 1, 2 ]
   }
}
```

Cauchy noise works well in terms of convergence and posterior mean estimation regardless of the value of σ_1^2. Conversely, normal noise only works for $\sigma_2^2 = 10$. For $\sigma_2^2 = 0.01$ and $\sigma_2^2 = 0.5$, convergence does not occur during the number of iterations or posterior mean approximation is too biased.



Exercise 4: Let X and Y be discrete random variables with support $\{x_1, \dots, x_n\}$ and $\{y_1, \dots, y_m\}$. Denote the joint probability mass function of X and Y by $p_{X,Y}(x, y) = \Pr(X = x, Y = y)$. Using a Gibbs sampler, assume that convergence to the distribution of (X, Y) has occurred. Demonstrate that the next state (x_{t+1}, y_{t+1}) will also be drawn from the same distribution as (X, Y).

Solution: By the law of total probability,

$$\Pr(X_{t+1} = x_{t+1}, Y_{t+1} = y_{t+1}) = \sum_{i,j} \Big[\Pr(X_{t+1} = x_{t+1}, Y_{t+1} = y_{t+1} \,|\, X_t = x_i, Y_t = y_j) \\ \Pr(X_t = x_i, Y_t = y_j) \Big] .$$

Assume that the next state (x_{t+1}, y_{t+1}) is generated by first drawing from the conditional distribution $Y \,|\, X$ and subsequently from $X \,|\, Y$. In that case

$$\Pr(X_{t+1} = x_{t+1}, Y_{t+1} = y_{t+1} \,|\, X_t = x_i, Y_t = y_j) = \Pr(X_{t+1} = x_{t+1} \,|\, Y_{t+1} = y_{t+1}) \times \\ \Pr(Y_{t+1} = y_{t+1} \,|\, X_t = x_i) .$$

The joint probability that $X_{t+1} = x_{t+1}$ and $Y_{t+1} = y_{t+1}$ is therefore

$$\begin{aligned} \Pr(X_{t+1} = x_{t+1}, Y_{t+1} = y_{t+1}) &= \Pr(X_{t+1} = x_{t+1} \,|\, Y_{t+1} = y_{t+1}) \times \\ &\quad \sum_{i,j} \Pr(Y_{t+1} = y_{t+1} \,|\, X_t = x_i) \Pr(X_t = x_i, Y_t = y_j) \\ &= \frac{p(x_{t+1}, y_{t+1})}{p(y_{t+1})} \sum_{i,j} \frac{p(x_i, y_{t+1})}{p(x_i)} p(y_j \,|\, x_i) p(x_i) \\ &= \frac{p(x_{t+1}, y_{t+1})}{p(y_{t+1})} \sum_{i} p(x_i, y_{t+1}) \sum_{j} p(y_j \,|\, x_i) \\ &= p(x_{t+1}, y_{t+1}) , \end{aligned}$$

which shows that (X_{t+1}, Y_{t+1}) follows the same distribution as (X, Y).

Exercise 5 (chapter 7.5): Let the vector $\boldsymbol{X} = [X_1, X_2]^{\mathrm{T}}$ follow a bivariate Normal distribution with zero mean vector and covariance matrix $\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ with $|\rho| < 1$.

1. Implement Monte Carlo simulation and Gibbs sampling to compute marginal expectations and variances.
2. Use $\rho = 0$ and generate 500 samples. Compare both methods in terms of bias.
3. Use $\rho = 0.5, 0.9, 0.99, 0.999$ and generate again 500 samples. Create trace plots and explain how the correlation affects Gibbs sampling.
4. Repeat 2. and 3. by generating 10 000 samples. Explain how Gibbs sampling improves in terms of bias when generating more samples.

Solution: The conditional density of X_1 given $X_2 = x_2$ is

$$
\begin{aligned}
f_{X_1|X_2}(x_1 \mid x_2) &= \frac{f_{X_1,X_2}(x_1, x_2)}{f_{X_2}(x_2)} \\
&= \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{-\frac{x_1^2 - 2\rho x_1 x_2 + x_2^2}{2(1-\rho^2)}\right\} \Big/ \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x_2^2}{2}\right\}, \\
&= \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp\left\{-\frac{(x_1 - \rho x_2)^2}{2(1-\rho^2)}\right\}
\end{aligned}
$$

which can be recognized as the density of a $\text{Normal}\big(\rho x_2, 1-\rho^2\big)$ distribution. Inversely, the conditional distribution of X_2 given $X_1 = x_1$ is $\text{Normal}\big(\rho x_1, 1-\rho^2\big)$. An implementation of Monte Carlo simulation and Gibbs sampling in R is:

```
rho <- 0.5 ; Sigma <- matrix( c( 1, rho, rho, 1 ), 2, 2 ) ;
scale <- sqrt( 1 - rho ^ 2 ) ; nSamples <- 10000 ;
x <- array( 0, c( nSamples, 2, 2 ) )
x[, , 1 ] <- mvtnorm::rmvnorm( nSamples, sigma = Sigma )
for( ii in seq( 2, nSamples ) ) {
   x[ ii, 1, 2 ] <- rnorm( 1, rho * x[ ii - 1, 2, 2 ], scale )
   x[ ii, 2, 2 ] <- rnorm( 1, rho * x[ ii - 1, 1, 2 ], scale )
}
```

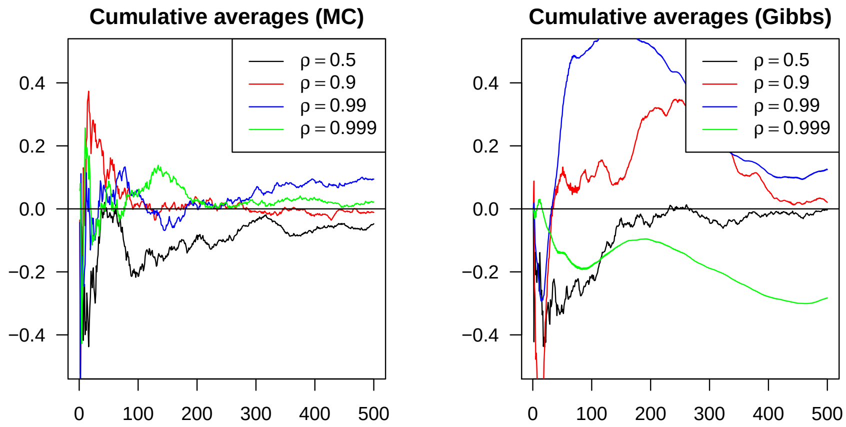
For $\rho = 0$ and 500 samples, the performance of both methods in terms of bias is:

```
# Expectations (MC): -0.01832 0.08145
# Expectations (Gibbs): -0.00554 0.06355
# Variances (MC): -0.06087 -0.07836
# Variances (Gibbs): 0.02799 0.07053
```

Monte Carlo simulation and Gibbs sampling are equivalent for $\rho = 0$, because the conditional distributions reduce to marginals and generating from the bivariate distribution is equivalent to drawing from the marginals due to independence. For $\rho = 0.5, 0.9, 0.99, 0.999$, the performance of both methods is:

```
#       E[X1] (MC) E[X2] (MC) Var[X1] (MC) Var[X2] (MC)
# 0.5     -0.04728   -0.05681       1.0927       1.0393
# 0.9     -0.01116   -0.01935       0.9589       0.9952
# 0.99     0.09452    0.09711       0.9814       0.9792
# 0.999    0.02218    0.02219       0.9756       0.9757

#       E[X1] (Gibbs) E[X2] (Gibbs) Var[X1] (Gibbs) Var[X2] (Gibbs)
# 0.5       -0.003253       0.05215          0.9698          1.0500
# 0.9        0.020826       0.01506          0.8167          0.8190
# 0.99       0.124466       0.13561          0.4741          0.4731
# 0.999     -0.282552      -0.28126          0.0935          0.0936
```



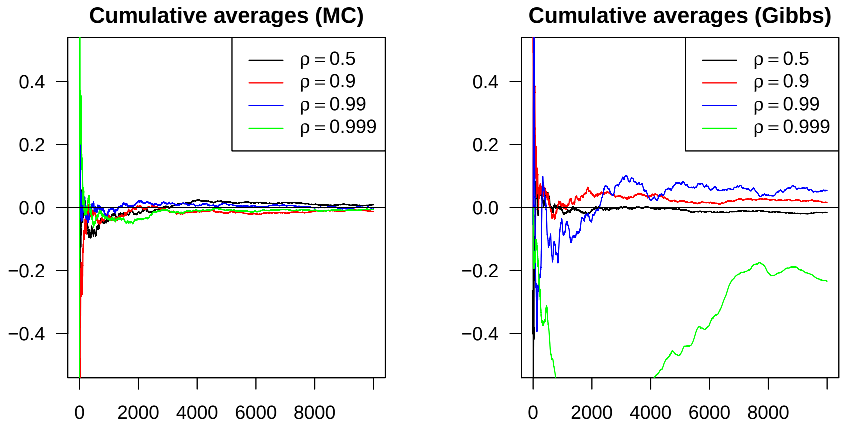
The performance of the Gibbs sampler in terms of bias decreases as the correlation ρ between X_1 and X_2 gets larger. Note the underestimation of the variances for large values of ρ. The decrease in accuracy is due to the large correlation betweens subsequent Gibbs draws. This behavior can be seen from the cumulative average plot: a large value of ρ results in a smooth graph.
For $\rho = 0$ and 10000 samples, the performance of both methods in terms of bias is:

```
# Expectations (MC): 0.0122 0.007333
# Expectations (Gibbs): -0.01249 -0.006557
# Variances (MC): -0.01316 0.008621
# Variances (Gibbs): 0.01814 0.008486
```

For $\rho = 0.5, 0.9, 0.99, 0.999$, the performance of both methods is:

```
#       E[X1] (MC) E[X2] (MC) Var[X1] (MC) Var[X2] (MC)
# 0.5     0.001906  -0.004195        1.001       0.9984
# 0.9    -0.003444  -0.001229        1.004       1.0026
# 0.99   -0.009029  -0.008474        1.003       1.0048
# 0.999   0.002127   0.001981        1.008       1.0089

#       E[X1] (Gibbs) E[X2] (Gibbs) Var[X1] (Gibbs) Var[X2] (Gibbs)
# 0.5        0.002819      0.001682          1.0080          1.0056
# 0.9        0.013091      0.012947          1.0093          1.0095
# 0.99       0.055644      0.055629          1.0287          1.0289
# 0.999     -0.234236     -0.234031          0.7897          0.7898
```



The performance of both methods increases with the number of generated draws. However, for very large values of ρ, Gibbs sampling is still very biased and convergence diagnostics are necessary.

Exercise 6 (chapter 7.5): Let $\{y\}_{i=1}^{n}$ be observations from a counting process where

$$y_i \,|\, \mu_1, \mu_2, \lambda \sim \begin{cases} \text{Poisson}(\mu_1) & \text{if } i \leq \lambda \\ \text{Poisson}(\mu_2) & \text{if } i > \lambda \end{cases}$$

and λ denotes a changepoint. Let the priors be

$$\begin{aligned} \mu_1 &\sim \text{Gamma}(\alpha_1, \beta_1) \\ \mu_2 &\sim \text{Gamma}(\alpha_2, \beta_2) \qquad . \\ \lambda &\sim \text{Uniform}(1, 2, \ldots, n) \end{aligned}$$

1. Find the likelihood and joint posterior density for the changepoint model.
2. Find all full conditional densities to implement a Gibb sampler.
3. Use the Gibbs sampler and the following data to perform changepoint detection:

$$\begin{aligned} &4, 4, 3, 1, 3, 2, 1, 0, 11, 11, 12, 4, 4, 7, 9, 6, 9, 12, 13, 15, 12, 10, 10, 6, 6, 7, 12, 11, \\ &15, 5, 11, 8, 11, 7, 11, 12, 14, 12, 8, 11, 9, 10, 6, 14, 14, 8, 4, 7, 10, 3, 14, 10, 17, 7, \\ &16, 9, 12, 11, 7, 11, 5, 11, 13, 9, 7, 9, 7, 11, 12, 13, 6, 9, 10, 13, 8, 18, 6, 16, 8, 4, 16, \\ &8, 9, 5, 7, 9, 10, 11, 13, 12, 9, 11, 7, 9, 6, 7, 6, 11, 8, 5 \end{aligned}$$

Solution: The likelihood and posterior density are

$$\begin{aligned} p(\mu_1, \mu_2, \lambda \,|\, y) &\propto p(y \,|\, \mu_1, \mu_2, \lambda) p(\mu_1) p(\mu_2) p(\lambda) \\ &= \left[\prod_{i=1}^{\lambda} \mu_1^{y_i} \exp\{-\mu_1\} \prod_{i=\lambda+1}^{n} \mu_2^{y_i} \exp\{-\mu_2\} \right] \left[\mu_1^{\alpha_1 - 1} \exp\{-\beta_1 \mu_1\} \right] \left[\mu_2^{\alpha_2 - 1} \exp\{-\beta_2 \mu_2\} \right] \end{aligned}$$

The full conditional density of μ_1 is

$$
\begin{aligned}
p(\mu_1 \,|\, \mu_2, \lambda, y) &\propto \prod_{i=1}^{\lambda} \mu_1^{y_i} \exp\{-\mu_1\} \mu_1^{\alpha_1 - 1} \exp\{-\beta_1 \mu_1\} \\
&= \mu_1^{\alpha_1 + (\sum_{i=1}^{\lambda} y_i) - 1} \exp\{-\mu_1(\beta_1 + \lambda)\}\,,
\end{aligned}
$$

which can be recognized as the density of a $\text{Gamma}\Big(\alpha_1 + \sum_{i=1}^{\lambda} y_i, \beta_1 + \lambda\Big)$ distribution. The full conditional density of μ_2 is

$$
\begin{aligned}
p(\mu_2 \,|\, \mu_1, \lambda, y) &\propto \prod_{i=\lambda+1}^{n} \mu_2^{y_i} \exp\{-\mu_2\} \mu_2^{\alpha_2 - 1} \exp\{-\beta_2 \mu_2\} \\
&= \mu_2^{\alpha_2 + (\sum_{i=\lambda+1}^{n} y_i) - 1} \exp\{-\mu_2(\beta_2 + n - \lambda)\}\,,
\end{aligned}
$$

which can be recognized as the density of a $\text{Gamma}\big(\alpha_2 + \sum_{i=\lambda+1}^{n} y_i, \beta_2 + n - \lambda\big)$ distribution. The full conditional (probability mass function) of λ is

$$
p(\lambda \,|\, \mu_1, \mu_2, y) \propto \prod_{i=1}^{\lambda} \mu_1^{y_i} \exp\{-\mu_1\} \prod_{i=\lambda+1}^{n} \mu_2^{y_i} \exp\{-\mu_2\} \qquad \lambda = 1, 2, \ldots, n\,.
$$

An implementation of Monte Carlo simulation and Gibbs sampling in R is:

```
a1 <- b1 <- a2 <- b2 <- 1
n <- length( y ) ; nSamples <- 2000
x <- matrix( 0, nSamples, 3 ) ; x[ 1, 3 ] <- 10
grid <- seq_len( n )
for( ii in seq( 2, nSamples ) ) {
   x[ ii, 1 ] <- rgamma(
      1,
      a1 + sum( y[ 1 : x[ ii - 1, 3 ] ] ),
      b1 + x[ ii - 1, 3 ]
   )
   x[ ii, 2 ] <- rgamma(
      1,
      a2 + sum( y[ ( x[ ii - 1, 3 ] + 1 ) : n ] ),
      b2 + n - x[ ii - 1, 3 ]
   )
   like1 <- cumsum( dpois( y[ grid ], x[ ii, 1 ], TRUE ) )
   like2 <- dpois( y[ grid[ -1 ] ], x[ ii, 2 ], TRUE )
   like2 <- sapply( 1 : length( like2 ), function( ii, nLike2 ) {
      sum( like2[ ii : nLike2 ] ), nLike2 = length( like2 )
   } )
   probs <- like1 + c( like2, 0 )
   maxProb <- max( probs )
```

```
    sumProbs <- maxProb + log( sum( exp( probs - maxProb ) ) )
    probs <- exp( probs - sumProbs )
    x[ ii, 3 ] <- sample( grid , 1, FALSE, probs )
}
```

Gibbs sampling using 10 000 draws resulted the following posterior mean estimates for the above data:

```
#    mu1    mu2 lambda
#  2.110  9.524  8.000
```

The true parameter values are $\mu_1 = 2$, $\mu_2 = 10$ and $\lambda = 8$.

Exercise 7 (chapter 7.8): Let $\{X_i\}_{i=1}^n$ be correlated random variables with $\mathbb{V}[X_i] = \sigma^2$ for all $i = 1, \ldots n$ and $\mathrm{Cov}[X_i, X_{i+k}] = \sigma_k$ for all i, k. Consider the sample mean $\bar{X} = n^{-1} \sum_{i=1}^n X_i$ and find its variance $\mathbb{V}\big[\bar{X}\big]$.

Solution: The variance of the sample mean is

$$
\begin{aligned}
\mathbb{V}\big[\bar{X}\big] &= \mathbb{V}\left[\frac{1}{n}\sum_{i=1}^n X_i\right] \\
&= \frac{1}{n^2}\mathbb{V}[X_1 + (X_2 + \ldots + X_n)] \\
&= \frac{1}{n^2}(\mathbb{V}[X_1] + 2\mathrm{Cov}[X_1, X_2 + \ldots + X_n] + \mathbb{V}[X_2 + \ldots + X_n]) \\
&= \frac{1}{n^2}\big(\sigma^2 + 2[\sigma_1 + \ldots + \sigma_{n-1}] + \mathbb{V}[X_2 + \ldots + X_n]\big)
\end{aligned} .
$$

Continuing in a similar manner with $\mathbb{V}[X_2 + \ldots + X_n]$ and all subsequent variances yields

$$
\begin{aligned}
\mathbb{V}\big[\bar{X}\big] &= \frac{1}{n^2}\big(n\sigma^2 + 2[(n-1)\sigma_1 + \ldots + \sigma_{n-1}]\big) \\
&= \frac{\sigma^2}{n}\left(1 + 2\left[\frac{(n-1)\sigma_1}{n\sigma^2} + \ldots + \frac{\sigma_{n-1}}{n\sigma^2}\right]\right) \\
&= \frac{\sigma^2}{n}\left(1 + 2\sum_{j=1}^{n-1}\left[\frac{n-j}{n}\right]\frac{\sigma_j}{\sigma^2}\right) \\
&= \frac{\sigma^2}{n}\left(1 + 2\sum_{j=1}^{n-1}\left[1 - \frac{j}{n}\right]\rho_j\right)
\end{aligned} ,
$$

where $\rho_j = \sigma_j/\sigma^2$ is the correlation at lag j, that is, the correlation between X_i and X_{i+k}. Note that the variance of the sample mean can be used in MCMC sampling to compute numerical standard errors. These numerical standard errors help quantify the uncertainty on $\mathbb{E}[h(\theta)]$ due to MCMC sampling. It can also be used to determine the number of MCMC draws as it tends to 0 with increasing draws.