1. (a) The contour plot for the log-likelihood, $\log L(\mu, \sigma | X_1, ..., X_{10})$.

(b) For independent prior, $\log g(\mu, \sigma^2) = \log g_1(\mu) + \log g_2(\sigma^2)$. When $\mu$ is fixed, $\log g(\mu, \sigma) = \log g_2(\sigma^2) + \text{constant}$ if $\mu$ and $\sigma$ are independent. The following figure shows their difference. The lines in the left hand side are parallel but the lines in the right hand side are not.
1. The contour plot for the log $g(\mu, \sigma)$ and log $L(\mu, \sigma) + \log g(\mu, \sigma)$ with Jeffreys improper priors, $g(\mu, \sigma) = \frac{1}{\sigma^2}$.

2. The contour plot for the log $g(\mu, \sigma)$ and log $L(\mu, \sigma) + \log g(\mu, \sigma)$ with a priori $g(\mu, \sigma)$, $\mu \sim \text{N}(0, 10^2)$ independent of $\sigma^2 \sim \text{Inv-}\chi^2(1, 10^2)$.

Please look at page 7 for more detailed contour plots of (b)(2)-(5).
3. The contour plot for the log \( g(\mu, \sigma) \) and log \( L(\mu, \sigma) + \log g(\mu, \sigma) \) with a priori \( g(\mu, \sigma), \mu \sim N(0, \sigma^2) \) independent of \( \sigma^2 \sim \text{Inv-\( \chi^2 \)}(3, \sigma^2) \).

4. The contour plot for the log \( g(\mu, \sigma) \) and log \( L(\mu, \sigma) + \log g(\mu, \sigma) \) with a priori \( g(\mu, \sigma), \mu \sim N(0, \sigma^2), \sigma^2 \sim \text{Inv-\( \chi^2 \)}(1, \sigma^2) \).

In the 1st to 3rd plot, the shape of marginal slices fixed and only location changed. In the 4th and 5th plot (next page), since the shape of marginal slices changed, the contour plots show fan shape in the right end.
5. The contour plot for the log $g(\mu, \sigma)$ and log $L(\mu, \sigma) + \log g(\mu, \sigma)$ with a priori $g(\mu, \sigma)$, $\mu \sim N(0, \sigma^2/10)$, $\sigma^2 \sim \text{Inv-}\chi^2(1, 2^2)$.

Several ways to find the pdf of a scaled Inv-$\chi^2$ distribution.

$X \sim \text{Inv-}\chi^2(\nu, s^2)$ with pdf $f(x) = \frac{(s^2 \nu/2)^{\nu/2}}{\nu^{\nu/2} \Gamma(\nu/2)} x^{\nu - 1} \exp\left(-\frac{s^2 \nu}{2x}\right)$, $\nu > 0, s^2 > 0, x > 0$

$\Leftrightarrow X \sim \text{Inv-}\Gamma\left(\frac{\nu}{2}, \frac{s^2}{2}\right)$

$\Leftrightarrow X = s^2 \frac{1}{\nu}, Y \sim \chi^2(\nu)$, get $f(x)$ by the transformation from $f(y)$

Rcode:

gsize=50
x<-c(4.90791,4.83425,5.19801,6.85308,4.07085,4.66076,4.16201,5.49752,4.15438,3.72703)
mu<-3+seq(-3,14,length=gsize)
sigma<-.1+seq(0,10,length=gsize)
theta<-expand.grid(mu,sigma)
logL<- function (theta){sum(dnorm(x,theta[1],theta[2], log=TRUE)) }
z<-apply(theta,1,logL)
w<-matrix(apply(theta,1, logG1),gsize,gsize)
contour(mu,sigma,w,main=main1 ,xlab="mu", ylab="sigma")
contour(mu,sigma,w+z,main=main2, xlab="mu", ylab="sigma")
# These will give a less detailed contour plot and will provide information to find better values of "level".
2 (a) The WinBUGS code specifies a set of bivariate Normal data $Y_i = (Y_{1i}, Y_{2i})^T \sim MVN(\mu, \Sigma = R^{-1} = D)$ for $i = 1, ..., 10$. The prior of $\mu$ is $\mu \sim MVN(\text{Alpha} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_0 = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix})$. (i.e. $\tau = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix}$). The prior of $\Sigma$ is $\Sigma = D = R^{-1} \sim \text{Inv-Wishart}_{\nu=4} \left( \text{Lambda}^{-1} = \Delta = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix} \right)$. Therefore, $\Sigma$ has mean $(\nu - k - 1)^{-1}\text{Lambda} = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$.

The following are statistic result from WinBUGS for 5 parameters, with 4 chains, initial values are generated by default, and beg from 90000 to end 100000 with thin 10 to get 4000 samples.

<table>
<thead>
<tr>
<th>node</th>
<th>mean</th>
<th>sd</th>
<th>MC error</th>
<th>2.5%</th>
<th>median</th>
<th>97.5%</th>
<th>start</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>mu[1]</td>
<td>4.892</td>
<td>0.5352</td>
<td>0.009465</td>
<td>3.865</td>
<td>4.877</td>
<td>6.029</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>mu[2]</td>
<td>4.254</td>
<td>0.5776</td>
<td>0.009552</td>
<td>3.04</td>
<td>4.273</td>
<td>5.384</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>rho</td>
<td>0.5822</td>
<td>0.2813</td>
<td>0.004229</td>
<td>-0.171</td>
<td>0.6583</td>
<td>0.9181</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>sig1</td>
<td>1.43</td>
<td>0.4093</td>
<td>0.002333</td>
<td>0.8715</td>
<td>1.352</td>
<td>2.425</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>sig2</td>
<td>1.626</td>
<td>0.4085</td>
<td>0.002777</td>
<td>1.048</td>
<td>1.56</td>
<td>2.63</td>
<td>90000</td>
<td>4000</td>
</tr>
</tbody>
</table>

(b) The following are statistic result from WinBUGS for 5 parameters, set $\nu = 20$ and $\text{Lambda} = \begin{pmatrix} 68 & 0 \\ 0 & 68 \end{pmatrix}$ such that hold the same prior mean for $\Sigma$, with 4 chains, initial values are generated by default, and beg from 90000 to end 100000 with thin 10 to get 4000 samples.

<table>
<thead>
<tr>
<th>node</th>
<th>mean</th>
<th>sd</th>
<th>MC error</th>
<th>2.5%</th>
<th>median</th>
<th>97.5%</th>
<th>start</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>mu[1]</td>
<td>4.664</td>
<td>0.6907</td>
<td>0.0108</td>
<td>3.327</td>
<td>4.666</td>
<td>6.077</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>mu[2]</td>
<td>4.456</td>
<td>0.6725</td>
<td>0.0118</td>
<td>3.111</td>
<td>4.457</td>
<td>5.759</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>rho</td>
<td>0.06784</td>
<td>0.1975</td>
<td>0.002921</td>
<td>-0.3219</td>
<td>0.07103</td>
<td>0.4437</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>sig1</td>
<td>1.823</td>
<td>0.2731</td>
<td>0.004557</td>
<td>1.38</td>
<td>1.794</td>
<td>2.43</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>sig2</td>
<td>1.871</td>
<td>0.2726</td>
<td>0.004557</td>
<td>1.423</td>
<td>1.837</td>
<td>2.485</td>
<td>90000</td>
<td>4000</td>
</tr>
</tbody>
</table>

Since the prior suggest higher $\sigma$, and the degrees of freedom $\nu = 20$ is increased to hold the same prior mean for $\Sigma$ when changing the $\text{Lambda}$, that induced higher mean of $\sigma$ and smaller $\rho$ and also induced wider credible intervals for $mu$, narrower credible intervals for $\sigma$ and smaller $\rho$.

(c) The following are statistic result from WinBUGS for 5 parameters, set $\Sigma_0 = \tau_0 = I_{2 \times 2}$, with 4 chains, initial values are generated by default, and beg from 90000 to end 100000 with thin 10 to get 4000 samples.

<table>
<thead>
<tr>
<th>node</th>
<th>mean</th>
<th>sd</th>
<th>MC error</th>
<th>2.5%</th>
<th>median</th>
<th>97.5%</th>
<th>start</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>mu[1]</td>
<td>2.257</td>
<td>0.9878</td>
<td>0.01849</td>
<td>0.1471</td>
<td>2.318</td>
<td>3.971</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>mu[2]</td>
<td>1.414</td>
<td>0.9284</td>
<td>0.01534</td>
<td>-0.4374</td>
<td>1.413</td>
<td>3.205</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>rho</td>
<td>0.8887</td>
<td>0.1214</td>
<td>0.002136</td>
<td>0.5471</td>
<td>0.9276</td>
<td>0.9855</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>sig1</td>
<td>3.061</td>
<td>1.257</td>
<td>0.02328</td>
<td>1.276</td>
<td>2.842</td>
<td>6.134</td>
<td>90000</td>
<td>4000</td>
</tr>
<tr>
<td>sig2</td>
<td>3.071</td>
<td>1.001</td>
<td>0.01477</td>
<td>1.573</td>
<td>2.917</td>
<td>5.451</td>
<td>90000</td>
<td>4000</td>
</tr>
</tbody>
</table>
Since the prior for $\mu$ suggest a more condensed distribution, the posterior means are closer to the prior means zero. That also increases the standard deviation and induce a wider credible interval.

(d)

stats result from WinBUGS for 5 parameters, set $\Lambda = \begin{pmatrix} 4 & -3 \\ -3 & 4 \end{pmatrix}$, with 4 chains, initial values are generated by default, and beg from 90000 to end 100000 with thin 10 to get 4000 samples.

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
\text{node} & \text{mean} & \text{sd} & \text{MC error} & \text{2.5%} & \text{median} & \text{97.5%} & \text{start} & \text{sample} \\
\hline
\mu[1] & 4.748 & 0.5364 & 0.09367 & 3.703 & 4.732 & 5.837 & 90000 & 4000 \\
\mu[2] & 4.383 & 0.5922 & 0.09807 & 3.153 & 4.397 & 5.548 & 90000 & 4000 \\
rho & 0.2609 & 0.3664 & 0.005364 & -0.5033 & 0.3024 & 0.8277 & 90000 & 4000 \\
\sigma[1] & 1.379 & 0.3938 & 0.00565 & 0.8584 & 1.295 & 2.378 & 90000 & 4000 \\
\sigma[2] & 1.612 & 0.4059 & 0.006606 & 1.038 & 1.539 & 2.621 & 90000 & 4000 \\
\hline
\end{array}
\]

Comparing to (b), this prior suggest a negative correlation between 2 variables. The posterior distribution reflects this change.

(e)

stats result from WinBUGS for 5 parameters, set $\Lambda = \begin{pmatrix} 40 & 0 \\ 0 & 40 \end{pmatrix}$, with 4 chains, initial values are generated by default, and beg from 90000 to end 100000 with thin 10 to get 4000 samples.

\[
\begin{array}{|c|c|c|c|c|c|c|c|}
\hline
\text{node} & \text{mean} & \text{sd} & \text{MC error} & \text{2.5%} & \text{median} & \text{97.5%} & \text{start} & \text{sample} \\
\hline
\mu[1] & 4.536 & 0.9746 & 0.01672 & 2.644 & 4.643 & 6.634 & 90000 & 4000 \\
\mu[2] & 4.432 & 0.9329 & 0.01547 & 2.527 & 4.443 & 6.297 & 90000 & 4000 \\
rho & 0.09748 & 0.3043 & 0.004519 & -0.4965 & 0.1126 & 0.6533 & 90000 & 4000 \\
\sigma[1] & 2.55 & 0.6948 & 0.009826 & 1.622 & 2.412 & 4.342 & 90000 & 4000 \\
\sigma[2] & 2.584 & 0.6362 & 0.01011 & 1.656 & 2.473 & 4.15 & 90000 & 4000 \\
\hline
\end{array}
\]

Compare to (b) the prior of $\sigma$ suggest higher expectation. The posterior distribution reflects this change.

The following table shows the estimate of means and covariance structure and the corresponding credible intervals and length of these intervals.

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|}
\hline
\text{Note} & \text{(a)Mean} & 2.50 & 97.50 & \text{Length} & \text{(b)Mean} & 2.50 & 97.50 & \text{Length} & \text{(c)Mean} & 2.50 & 97.50 & \text{Length} \\
\hline
rho & 0.5822 & -0.171 & 0.9181 & 1.0891 & 0.06784 \uparrow & -0.3219 & 0.4437 & 0.7656 & 0.8887 \uparrow & 0.5471 & 0.9855 & 0.4384 \\
\sigma[1] & 1.43 & 0.8715 & 2.425 & 1.5535 & 1.823 & 1.38 & 2.43 & 1.05 & 3.061 \uparrow & 1.276 & 6.134 & 4.858 \\
\hline
\text{Note} & \text{(d)Mean} & 2.50 & 97.50 & \text{Length} & \text{(e)Mean} & 2.50 & 97.50 & \text{Length} \\
\hline
rho & 0.2699 \downarrow & -0.5033 & 0.8277 & 1.331 & 0.09748 \downarrow & -0.4965 & 0.6533 & 1.1498 \\
\sigma[1] & 1.379 & 0.8584 & 2.378 & 1.5196 & 2.55 & 1.622 & 4.342 & 2.72 \\
\sigma[2] & 1.612 & 1.038 & 2.621 & 1.583 & 2.584 & 1.656 & 4.15 & 2.494 \\
\hline
\end{array}
\]
1(b) Complement (the supports of $\mu$ and $\sigma$ from the original keys are not wide enough)
3. With \( k = 3 \), \( D = \Delta = \text{diag}(1,1,10) \), the following OpenBUGS/WinBUGS code could sample from \( \text{Inverse-Wishart} \) \((\nu = k + 1 = 4, \Lambda = \Delta^{-1} = \text{diag}(10,1,1)) \).

```plaintext
#########################################################################
model {
  R[1:3 , 1:3] ~ dwish(Lambda[,], nu)
  Sigma[1:3 , 1:3] <- inverse(R[1:3 , 1:3])
  diag1 <- Sigma[1,1]
  diag2 <- Sigma[2,2]
  diag3 <- Sigma[3,3]
  rho12 <- Sigma[1,2] / sqrt(diag1 * diag2)
  rho13 <- Sigma[1,3] / sqrt(diag1 * diag3)
  rho23 <- Sigma[2,3] / sqrt(diag2 * diag3)
}
list(nu=4,
     Lambda = structure(.Data = c(10,0,0,0,1,0,0,0,.1), .Dim = c(3, 3))
)
#########################################################################
```

The following are the statistic results from OpenBUGS with initial values generated by default, and beg from 10000 with thin 10 to get 50000 samples. Note that the medians of diagonal elements are 7.165, .7209, and .07149, which are very close to \( .72/\delta_{11} = 7.2 \), \( .72/\delta_{22} = .72 \), and \( .72/\delta_{33} = .072 \). The densities of correlations are flat in \((-1,1)\), agree with the claim that all correlations uniformly distributed on \((-1,1)\).
4. Error message shows up when running the given code. The updater cannot sample the node. This is not a probability model for $(Y, X)$, because

$$f(x|y) = \frac{1}{\sqrt{2\pi}}e^{-(x-y)^2/2}; \quad f(y) = c, \text{ where } c > 0 \text{ (Note that } f(y) \text{ is not a density);}$$

$$\int_{\mathbb{R}} \int_{\mathbb{R}} f(x,y) dx dy = \int_{\mathbb{R}} \int_{\mathbb{R}} f(x|y)f(y) dx dy = \int_{\mathbb{R}} \left[ \int_{\mathbb{R}} f(x|y) dx \right] f(y) dy = \int_{\mathbb{R}} f(y) dy = c \int_{\mathbb{R}} dy = \infty.$$ 

The Gibbs sampling algorithm for $f(x,y) \propto e^{-(x-y)^2/2}$

1. Start with $y^{(0)} = 0, x^{(0)} = 0$.
2. For $j = 1, 2, \ldots, N$, generate $y^{(j)} \sim N(x^{(j-1)}, 1)$; generate $x^{(j)} \sim N(y^{(j)}, 1)$.

Similar as question 2(c) in 2008 midterm exam, the sequence $y^{(1)}, x^{(1)}, y^{(2)}, x^{(2)}, \ldots$ is a normal random walk. It has variance $\text{Var}(y^{(n)}) = 2(n-1) \rightarrow \infty$. So the empirical distribution will not converge. Both marginals will increasingly spread out as $n \rightarrow \infty$.

Y = X = rep(NA, 1001)
Y[1] = X[1] = 0
for (j in 2:1001) {
  Y[j] <- rnorm(1,X[j-1],1)
  X[j] <- rnorm(1,Y[j],1)
}

Y = X = rep(NA, 1001)
Y[1] = X[1] = 0
for (j in 2:1001) {
  Y[j] <- rnorm(1,X[j-1],1)
  X[j] <- rnorm(1,Y[j],1)
}
5. **Metropolis algorithm** with proposal \( J(x|x') = \frac{1}{2\pi\sigma} e^{-(x-x')^2/(2\sigma^2)}; \)

(1) Start at initial point \( x^{(0)} = 0 \).
(2) For \( j = 1, 2, \ldots, N \),
   - Generate the candidate \( x^* \) from \( N(x^{(j-1)}, \sigma^2) \).
   - Calculate \( r^{(j)} = \frac{f(x^*)/J(x^*|x^{(j-1)})}{f(x^{(j-1)})/J(x^{(j-1)}|x^*)} = \frac{0.9\phi(x^*)+1.0\phi(x^*-10)}{0.9\phi(x^{(j-1)})+1.0\phi(x^{(j-1)}-10)} \).
   - Generate \( y^{(j)} \sim \text{Bernoulli}(\min(r^{(j)}, 1)) \).
   - Take \( x^{(j)} = y^{(j)} x^* + (1-y^{(j)}) x^{(j-1)} \).

With the following R code we can get the history plots for \( \sigma = 0.01, 1, 10, 100 \).

```r
# R code for Metropolis algorithm
sigma = c(.01,.1,1,10,100)
x = matrix(0,nrow=1001,ncol=5)
for (i in 1:5){
  for (j in 2:1001) {
    cand = rnorm(1, x[j-1,i], sigma[i])
    r = (0.9*dnorm(cand)+.1*dnorm(cand-10))/(0.9*dnorm(x[j-1,i])+.1*dnorm(x[j-1,i]-10))
    u = runif(1)
    x[j,i] = ifelse(u<r, cand, x[j-1,i])
  }
}
```

**Discussion of tuning \( \sigma \):** From the history plots above, we can see that as \( \sigma \) increases, the number of accepted candidates becomes less, i.e., the rejection rate increases. In this case it will take too long to figure out the transfer probability between states. However, if we unify the range of \( x \), as shown below, it is easy to see that when \( \sigma \) is small, the chain moves too slowly that it can not reach the other 'island' in 1000 iterations. To sum up, we neither want \( \sigma \) to be too big nor too small, the tuning of a Metropolis jumping distribution should be very careful.
6. (a) Gibbs algorithm with the conditional distributions $X_1|X_2 = x_2 \sim N(0.99x_2, 0.0199)$; $X_2|X_1 = x_1 \sim N(0.99x_1, 0.0199)$:

1. Start at the initial point $x_1^{(0)} = 0, x_2^{(0)} = 0$.
2. For $j = 1, 2, \ldots, N$,
   - sample $x_1^{(j)}$ from $N(0.99x_2^{(j-1)}, 0.0199)$;
   - sample $x_2^{(j)}$ from $N(0.99x_1^{(j)}, 0.0199)$.

`N = 1000
sigma = sqrt(.0199)
x = matrix(0,nrow=N+1,ncol=2)
for (j in 2:(N+1)) {
x[j,1] = rnorm(1,.99*x[j-1,2],sigma)
x[j,2] = rnorm(1,.99*x[j,1],sigma)
}

To compare the empirical distributions with iteration $N = 1000, 5000, 10000$, I draw the samples on a two-dimension plot, and the contour lines are from a theoretical bivariate normal with $\mu$ and $\Sigma$ defined in the question. We can see that when $N = 1000$ the points mostly scattered in the inner contour line. The shape of the distribution looks like a linear band instead of an ellipse, and the tails are unlike normal distribution. When $N$ increases to 5000, the shape looks more like an ellipse; while as $N = 10000$, the sample fits the theoretical contour lines well. If checking the marginal distribution of $X_1$ and $X_2$ using histogram or QQ plot, $N = 5000$ is better than 1000, and seems enough to “see” the Normal.
(b) Gibbs algorithm with the conditional distributions $Z_1|Z_2 \sim N(0, 1)$; $Z_2|Z_1 \sim N(0, 1)$:

1. Start at the initial point $z_1^{(0)} = 0$, $z_2^{(0)} = 0$.
2. For $j = 1, 2, \ldots, N$, sample $z_1^{(j)}$, $z_2^{(j)}$ from $N(0, 1)$.

Actually $Z_1$ and $Z_2$ are independent here. Using similar code as part (a) and taking $N = 100$, 500, 1000, I get the following plots. 100 iterations seem not enough to see the standard bivariate normal. $N = 500$ is probably an appropriate iteration number, because from the scatter plot the points are dense in the center and sparse outside with a circle shape, and the marginal distribution looks like Normal. With 1000 iteration the bivariate normal shape looks more clear.

For 1000 pairs of $(z_1, z_2)^T$, calculate

$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \sqrt{1.96} \\ \sqrt{1.96} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$.  

The empirical joint distribution and marginal distributions are shown below.
Note that the theoretical distribution of \((Y_1, Y_2)\) is the same as \((X_1, X_2)\) in part (a). To prove this, we can use the fact: if \(Y \sim AX\) where \(A\) is a known \(k \times k\) matrix and \(X \sim \text{MVN}_k(\mu, \Sigma)\), then \(Y \sim \text{MVN}_k(A\mu, A\Sigma A^T)\).

Hence it is easy to show
\[
\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \sim \text{MVN}_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.99 \\ 0.99 & 1 \end{pmatrix}\right).
\]

Compare the empirical distribution of \((Y_1, Y_2)\) with \((X_1, X_2)\) for \(N = 1000\). The following plots show that although the two methods sample from the same distribution, with same number of iterations, both using Gibbs algorithm, the empirical distribution of transformed standard Normal looks better than the distribution sampling directly from the original bivariate Normal. This is an issue we discussed in class. Sometimes it is slow to sample from the original distribution, so we could consider different parameterizations.

(c) Metropolis algorithm using proposal \(J\left((x_1, x_2)\right) = \begin{pmatrix} (x_1, x_2) \end{pmatrix} = \begin{pmatrix} x_1' \\ x_2' \end{pmatrix}\) to be the density of \(\text{MVN}_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.99 \\ 0.99 & 1 \end{pmatrix}\right)\).

1. Start at initial point \(\begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}\).

2. For \(j = 1, 2, \ldots, N\),

   - Generate the candidate \(\begin{pmatrix} x_1^* \\ x_2^* \end{pmatrix}\) from \(\text{MVN}_2\left(\begin{pmatrix} x_1^{(j-1)} \\ x_2^{(j-1)} \end{pmatrix}, \sigma I\right)\).

   - Calculate \(r^{(j)} = \frac{f(x_1^*, x_2^*)/J((x_1^*, x_2^*)|(x_1^{(j-1)}, x_2^{(j-1)}))}{f(x_1^{(j)}, x_2^{(j)})/J((x_1^{(j)}, x_2^{(j)})|(x_1^{(j-1)}, x_2^{(j-1)}))}\).

   - Generate \(y^{(j)} \sim \text{Bernoulli}(|r^{(j)}|, 1))\).
\[
\begin{pmatrix}
  x_1^{(j)} \\
  x_2^{(j)}
\end{pmatrix} = y^{(j)} \begin{pmatrix}
  x_1^* \\
  x_2^*
\end{pmatrix} + (1 - y^{(j)}) \begin{pmatrix}
  x_1^{(j-1)} \\
  x_2^{(j-1)}
\end{pmatrix}.
\]

Given

```r
library(LearnBayes)
N = 1000
ct = c(.01,.1,1,10)
mu = c(0,0)
sigma = matrix(c(1,.99,.99,1),nrow=2)
x = array(0,dim=c(N+1,2,length(ct)))
cand=c(NA,NA)
for (i in 1:length(ct)) {
  for (j in 2:(N+1)) {
    x[j,,i] = x[j-1,,i]
    cand[1] = rnorm(1,x[j-1,1,i],sqrt(ct[i]))
    cand[2] = rnorm(1,x[j-1,2,i],sqrt(ct[i]))
    r = dmnorm(cand,mu,sigma)/dmnorm(x[j-1,,i],mu,sigma)
    u = runif(1)
    if (u<r) {
      x[j,,i] = cand
    }
  }
}
```
Discussion of tuning $c$. From the history plots and the scatterplots for different $c$, we see that when $c$ is small (e.g., 0.01), the acceptance rate is high, but each step is so tiny that during the 1000 iterations both $X_1$ and $X_2$ only move within a relatively small area. However, if $c$ is large (e.g., 10), the steps are big, while the acceptance rate is too low that we only obtain a few different updates in 1000 iterations. Considering both the steps and acceptance rate, $c = .1$ is the best choice of the four.