MAST30027: Modern Applied Statistics

Week 12 Lab Sheet

1. The Poisson distribution has the probability density function (pdf)

$$f(x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \text{ for } x = 0, 1, \ldots$$

   (a) Show that the Poisson distribution is an exponential family, identifying the parameters θ and ϕ as well as the functions $b(\theta)$ and $a(\phi)$.

   (b) Obtain the canonical link. Show your work.

   (c) Obtain the variance function. Show your work.

Solution:

#1. (a) $\log f(x|\lambda) = \log \frac{\lambda^x e^{-\lambda}}{x!} = x \log \lambda - \lambda - \log x!$

[4] $\theta = \log \lambda$ $\phi = 1$

$$= \frac{x\theta - b(\theta)}{a(\phi)} - C(x, \phi)$$

$b(\theta) = \lambda = e^\theta$ $a(\phi) = \phi$ $C(x, \phi) = -\log x!$

This is in the form of an exponential family.

(b) $b'(\theta) = e^\theta = \mu$ $\theta = (b')^{-1}(\mu) = \log \mu = g(\mu)$

[3]

(c) $b'(\theta) = e^\theta = \mu$ $b''(\theta) = e^\theta$

[3] $V(\mu) = b''(\theta) = b''((b')^{-1}(\mu)) = b''(\log \mu)$

$= e^{\log \mu} = \mu$

2. Suppose that $Y \sim NB(r, p)$ for some known non-negative integer valued r and unknown p. That is

$$P(Y = y) = \binom{y + r - 1}{r - 1} (1 - p)^r p^y$$
$$E(Y) = \frac{pr}{1 - p}.$$

If p is the probability of success, then Y is the number of successful trials until the r-th failure. What are the log-likelihood, observed information, and Fisher information for this example?

Solution:

Log likelihood

$$\ell(p) = \log\binom{y+r-1}{r-1} + r\log(1-p) + y\log p$$

$$\frac{\partial}{\partial p}\ell(p) = -\frac{r}{1-p} + \frac{y}{p}$$

$$\frac{\partial^2}{\partial p^2}\ell(p) = \frac{-r}{(1-p)^2} - \frac{y}{p^2}$$

Observed information $\Rightarrow \dfrac{r}{(1-p)^2} + \dfrac{y}{p^2}$

Fisher's information

$$I(p) = E\left[\frac{r}{(1-p)^2} + \frac{y}{p^2}\right] = \frac{r}{(1-p)^2} + \frac{\cancel{p}r}{1-p}\cdot\frac{1}{p\cancel{2}}$$

$$= \frac{r}{1-p}\left(\frac{p+1-p}{(1-p)p}\right) = \frac{r}{p(1-p)^2}$$

3. Suppose that $y \sim N(\mu, 1)$. What is the Jeffreys prior for μ?

Solution:

$$y \sim N(\mu, 1)$$

$$l(\mu) = -\log\sqrt{2\pi} - \frac{(y-\mu)^2}{2}$$

$$\frac{\partial}{\partial\mu} l(\mu) = (y-\mu)$$

$$\frac{\partial^2}{\partial\mu^2} l(\mu) = -1$$

$$I(\mu) = 1$$

$$p(\mu) \propto \sqrt{I(\mu)} \propto 1$$

---

4. Consider a random sample $X_1, \cdots, X_n$ satisfying $X_i \stackrel{d}{=} \text{pois}(\theta)$, i.e., $f(x_i|\theta) = \frac{\theta^{x_i} e^{-\theta}}{x_i!}$. To assess an estimator $\hat{\theta} = t(X_1, \cdots, X_n)$ of θ we use the loss function $L(\hat{\theta}; \theta) = \dfrac{(\hat{\theta} - \theta)^2}{\theta}$. We assume a gamma prior pdf $\theta \stackrel{d}{=} \text{gamma}(\beta, \kappa)$ with known β and κ, i.e. $p(\theta) = \dfrac{1}{\beta^\kappa \Gamma(\kappa)} \theta^{\kappa-1} e^{-\theta/\beta}$; $\theta > 0$.

   (a) Show that the Bayes estimator under the given loss function is $\hat{\theta} = (E[\theta^{-1}|\mathbf{x}])^{-1}$.

   (b) What is the posterior distribution of θ. Show your work.

   (c) Find a closed form for $\hat{\theta}$ in (a) by using the result of (b). Show your work. [Hint: $\Gamma(n) = (n-1)!$].

Solution:

   (a) $E_\theta[R_{\hat{\theta}}(\theta)] = E_\theta[E_{\mathbf{X}}[\frac{(\hat{\theta}-\theta)^2}{\theta}|\theta]] = E_{\mathbf{X}}(E_\theta[\frac{(\hat{\theta}-\theta)^2}{\theta}|\mathbf{X}])$. Hence for each given sample $\mathbf{x}$, the Bayes estimator $\hat{\theta} = \arg\min_{\theta} E_\theta[\frac{(\hat{\theta}-\theta)^2}{\theta}|\mathbf{x}]$. Now $E_\theta[\frac{(\hat{\theta}-\theta)^2}{\theta}|\mathbf{x}] = E[\theta^{-1}|\mathbf{x}]\hat{\theta}^2 - 2\hat{\theta} + E[\theta|\mathbf{x}]$ which is a quadratic function of $\hat{\theta}$. It is easy to see that it is minimised when $\hat{\theta} = (E[\theta^{-1}|\mathbf{x}])^{-1}$ which gives the Bayes estimator of θ.

(b) $p(\theta \mid x_1 \ldots x_n) \propto p(x_1 \ldots x_n \mid \theta)\, p(\theta)$

[6]

$$\propto \prod_{i=1}^{n} \frac{\theta^{x_i} e^{-\theta}}{x_i!} \theta^{k-1} e^{-\frac{\theta}{\beta}}$$

$$\propto \theta^{\sum_{i=1}^{n} x_i + k - 1} e^{-(n+\frac{1}{\beta})\theta}$$

$$\sim \text{Gamma}\Big(\underbrace{\big(n+\tfrac{1}{\beta}\big)^{-1}}_{=\beta^*}, \underbrace{\sum_{i=1}^{n} x_i + k}_{=k^*}\Big)$$

(c) $E(\theta^{-1} \mid x) = \int \frac{1}{\theta} \frac{1}{\beta^{*k^*}\Gamma(k^*)} \theta^{k^*-1} e^{-\frac{\theta}{\beta^*}} d\theta$

[6]

$$= \frac{1}{\beta^{*k^*}\Gamma(k^*)} \int \boxed{\theta^{k^*-2} e^{-\frac{\theta}{\beta^*}}}\, d\theta$$

$$= \frac{1}{\beta^{*k^*}\Gamma(k^*)} \frac{\beta^{*k^*-1}\Gamma(k^*-1)}{1} \underbrace{\int \frac{1}{\beta^{*(k^*-1)}\Gamma(k^*-1)} \boxed{}\, d\theta}_{=1}$$

$$= \frac{1}{\beta^*} \frac{1}{(k^*-1)} = \left(n+\frac{1}{\beta}\right) \frac{1}{\sum_{i=1}^{n} x_i + k - 1}$$

$$\hat{\theta} = E(\theta^{-1} \mid x)^{-1} = \frac{\sum_{i=1}^{n} x_i + k - 1}{n+\frac{1}{\beta}}$$

5. An expert tells you that the growth rate of a population is between 0.2 and 0.5 with 90% probability. Choose a log-normal distribution to match these prior beliefs.

Solution: We want X such that $\mathbb{P}(X < 0.2) = \mathbb{P}(X > 0.5) = 0.05$ and $\log X$ is normal. That is $\mathbb{P}(\log X < -1.6094) = \mathbb{P}(\log X > -0.6931) = 0.05$. Since $Y = \log X$ is symmetric about its mean, we have $\mu = \mathbb{E}Y = (-1.6094 - 0.6931)/2 = -1.1513$, and $\sigma^2 = \operatorname{Var} Y$ is such that

$$\mathbb{P}\left(\frac{Y + 1.1513}{\sigma} < \frac{-1.6094 + 1.1513}{\sigma}\right) = 0.05$$

whence $\sigma = (-1.6094 + 1.1513)/z_{0.05} = 0.2785$.

6. **Metropolis-Hastings** [**We already solved problems (a), (b), and (c) in the week 10. This week, solve the problem (d), (e), and (f).**]

Recall that $\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ and $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}$, iff $\mathbf{X}$ has joint density

$$f_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right),$$

where $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$.

(a) Write an R function that evaluates the density of a bivariate normal distribution. The function should take as input the point $\mathbf{x}$, the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$.

You will find the functions `solve` and `det` useful.

Solution:

```
> dbinorm <- function(x, mu, Si) {
+   # x and mu are vectors length 2 and Si a 2x2 matric
+   # returns the density at x of a bivariate normal mean mu var Si
+   exp(-t(x - mu)%*%solve(Si, x - mu)/2)/2/pi/sqrt(det(Si))
+ }
```

You can check that it is working by noting that `dbinorm(c(1,1), c(0,0), matrix(c(1,0,0,1),2,2))` gives the same answer as `dnorm(1)^2`.

(b) Write a program in R that uses the Metropolis-Hastings algorithm to generate a sample of size $n = 1000$ from the $N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}\right)$ distribution. Use the symmetric random walk proposal distribution $N\left(\mathbf{x}, \sigma^2 I\right)$ with $\sigma = 2.5$.

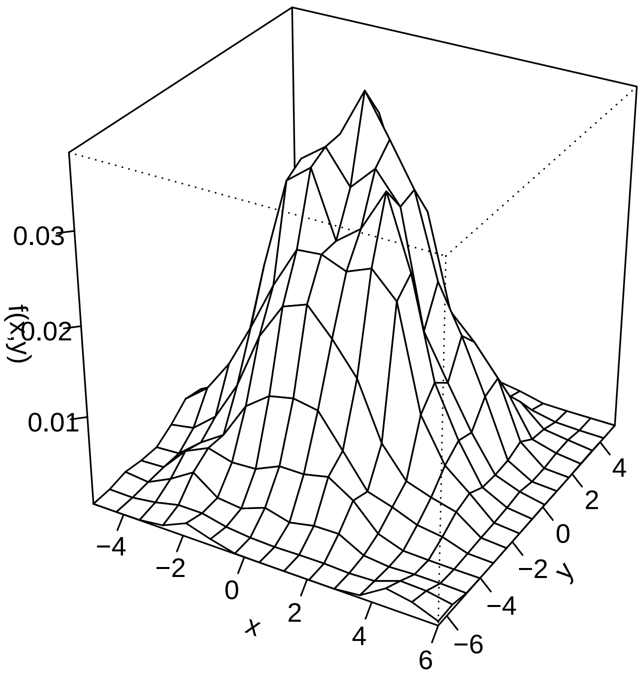
Use $\mathbf{X}(0) = \begin{pmatrix} 6 \\ -6 \end{pmatrix}$ as your initial state. Report the proportion of accepted values.

Solution: The acceptance rate was 46% (this will vary a little every time you run the program). To check that the output looks ok, I have plotted (a kernel density estimate of) the joint density. Easier than plotting a joint density would be to plot histograms/densities of the marginal samples, using `hist` or `density`.

```
> # Metropolis-Hastings simulation of a bivariate normal
> # inputs
> mu <- c(0, 0)                        # mean
> Si <- matrix(c(4, 1, 1, 4), 2, 2) # variance
> iterations <- 1000        # sample size
> startvalue <- c(6, -6)    # initial value
> sd <- 2.5                 # std-dev for proposal chain
> # main loop
> chain <- matrix(nrow = iterations+1, ncol = 2)
> chain[1,] <- startvalue
> accepted <- 0 # counts num accepted proposals
> for (i in 1:iterations){
+   proposal <- rnorm(2, chain[i,], sd)
+   prob <- dbinorm(proposal, mu, Si)/dbinorm(chain[i,], mu, Si)
+   if (is.nan(prob)) prob <- 0
+   if (runif(1) < prob) {
+     chain[i+1,] <- proposal
+     accepted <- accepted + 1
+   } else {
+     chain[i+1,] <- chain[i,]
+   }
+ }
> # acceptance rate
> accepted/iterations
```

```
[1] 0.457
```

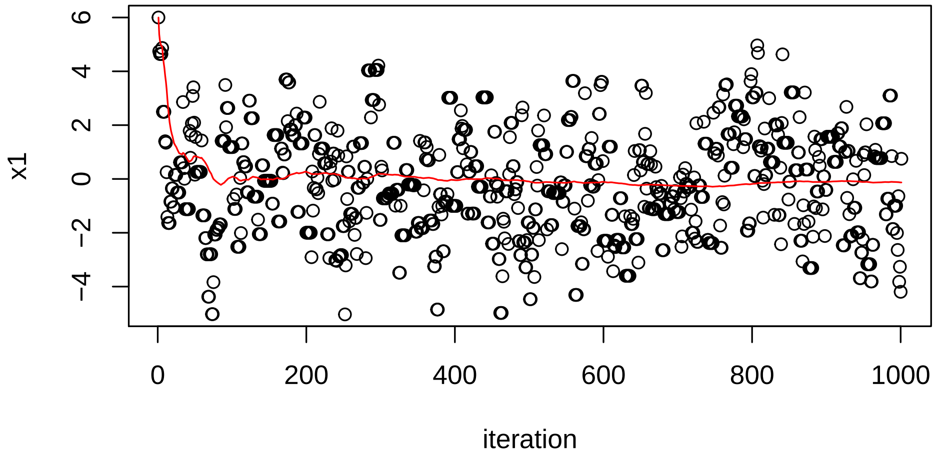
```
> # 2D density plot
> library(MASS)
> chain_density <- kde2d(chain[,1], chain[,2], n = 15)
> persp(chain_density, phi = 30, theta = 30, d = 5,
+       xlab = "x", ylab = "y", zlab = "f(x,y)",
+       ticktype = "detailed")
```

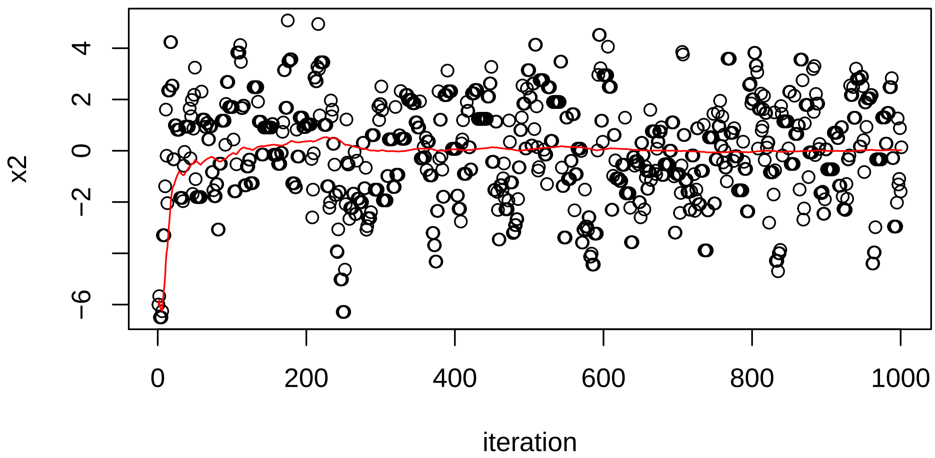


(c) Let $\mathbf{X}(n)$ be the n-th sample point. Plot $X_i(n)$ and the cumulative averages $\overline{X}_i(n) = n^{-1}\sum_{j=1}^{n} X_i(j)$, for $i = 1, 2$. The cumulative averages should give a rough idea of how quickly the $\mathbf{X}(n)$ converge in distribution.

Solution:

```
> # cumulative averages
> par(mfrow=c(2,1), mar=c(4,4,1,1))
> plot(chain[,1], xlab = "iteration", ylab = "x1")
> cumavg1 <- cumsum(chain[,1])/1:(iterations + 1)
> lines(1:(iterations + 1), cumavg1, col = "red")
> plot(chain[,2], xlab = "iteration", ylab = "x2")
> cumavg2 <- cumsum(chain[,2])/1:(iterations + 1)
> lines(1:(iterations + 1), cumavg2, col = "red")
```





(d) You can use the R functions `cor` or `acf` to estimate the lag 1 autocorrelation ρ. Calculate the so called *effective sample size*: $n(1-\rho)/(1+\rho)$ for each variable.

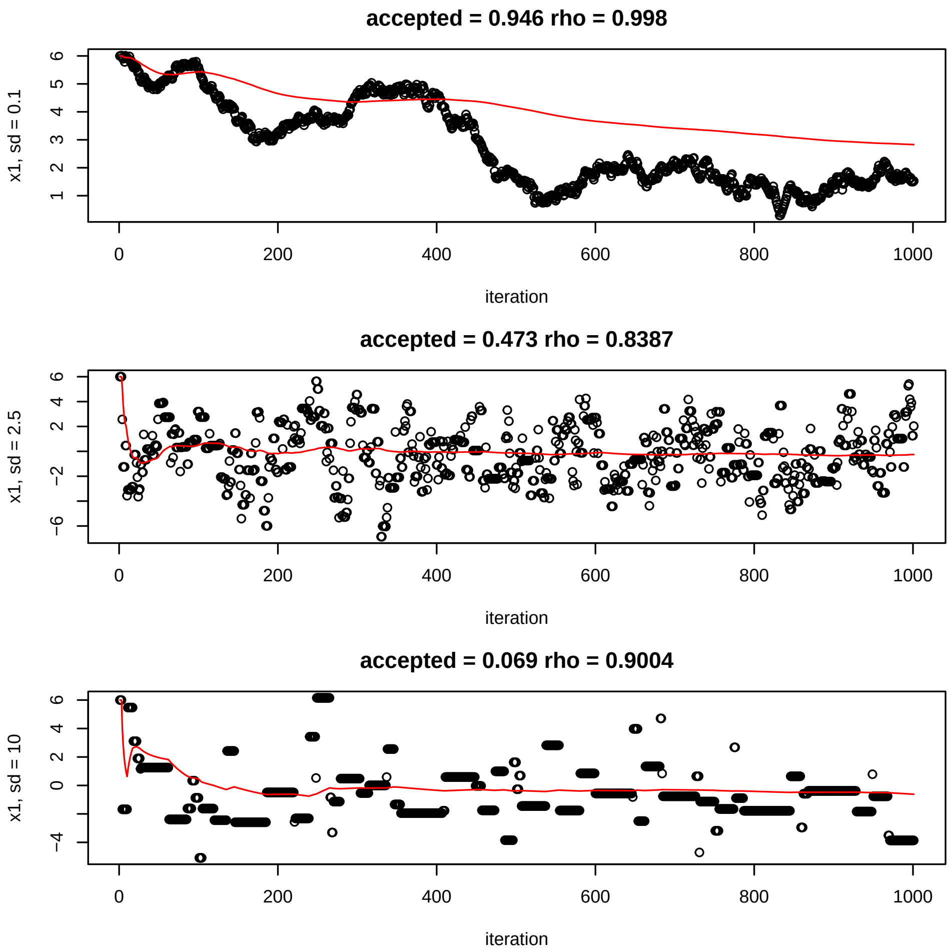
Solution: For both chains the lag 1 autocorrelation was quite high, at 75% and 79%. This is influenced by both the acceptance rate (lower rate means higher correlation) and the size of steps made by the proposal chain (smaller steps means higher correlation, but smaller steps also generally mean a higher acceptance rate). The effective sample sizes were 144 and 115 (out of 1000).

```
> # effective sample size
> (rho1 <- cor(chain[-(iterations+1),1], chain[-1,1]))

[1] 0.7475516

> rho1

[1] 0.7475516

> iterations * (1 - rho1)/(1 + rho1)

[1] 144.4583

> (rho2 <- cor(chain[-(iterations+1),2], chain[-1,2]))

[1] 0.7931476

> rho2

[1] 0.7931476

> iterations * (1 - rho2)/(1 + rho2)

[1] 115.3571
```

(e) Change the standard deviation of the proposal chain to $\sigma = 0.1$ and then $\sigma = 10$. How do the proportion of accepted values, the cumulative averages, and the effective sample size change?

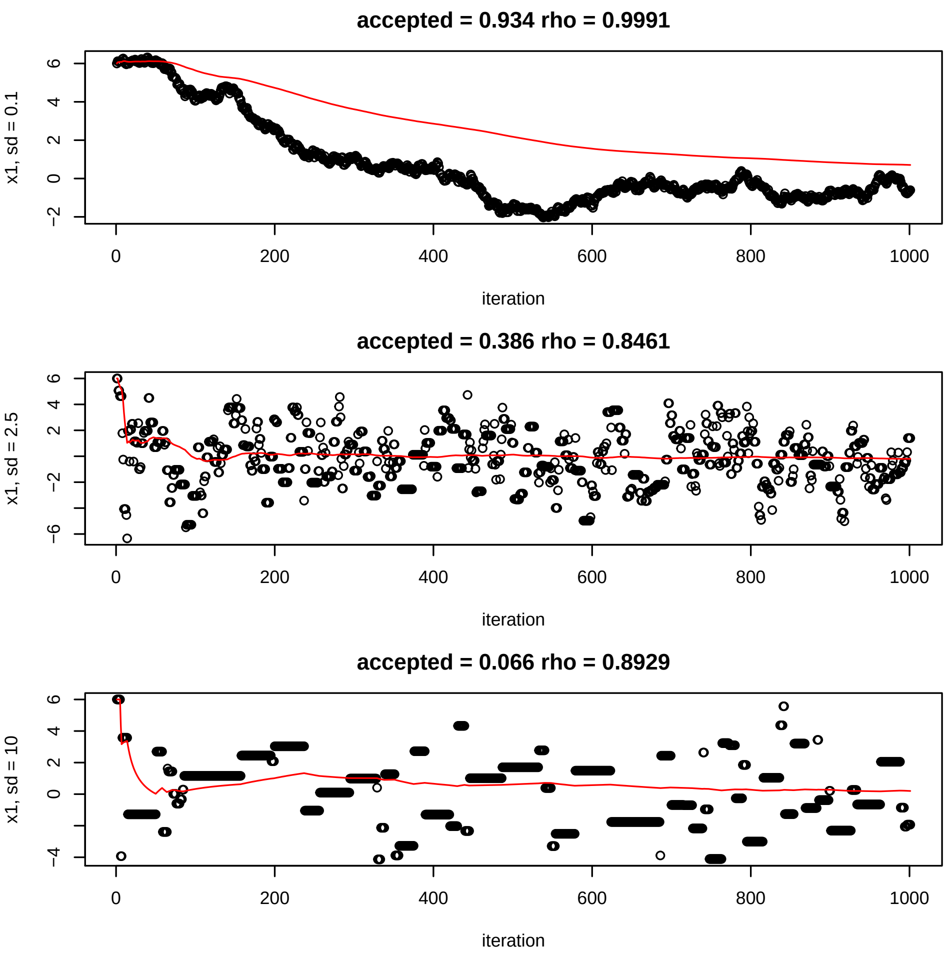
Solution: We just give the results for the first co-ordinate. The second co-ordinate behaves similarly.

We see that when $\sigma = 0.1$ the steps of the proposal chain are too small, meaning that the MCMC chain moves too slowly, resulting in a very high autocorrelation (and correspondingly small effective sample size). Conversely when $\sigma = 10$ the MCMC chain moves slowly because so few proposal steps are accepted, again resulting in a high autocorrelation. Our original MCMC chain, with $\sigma = 2.5$, seems to be a happy medium.



(f) Now change $\mathbf{\Sigma}$ to $\begin{pmatrix} 4 & 2.8 \\ 2.8 & 4 \end{pmatrix}$ and repeat the above analysis. How do things change?

Solution: Greater correlation between X_1 and X_2 means that the target chain is less like the proposal chain. This decreases the acceptance rate and hence tends to increase the autocorrelation (in the third case there is actually a tiny decrease).


accepted = 0.934 rho = 0.9991
x1, sd = 0.1
iteration
accepted = 0.386 rho = 0.8461
x1, sd = 2.5
iteration
accepted = 0.066 rho = 0.8929
x1, sd = 10
iteration