

ST407-Monte Carlo Methods

Assignment 2

Name: Christis Katsouris
Student ID: 1154579
Department of Statistics
University of Warwick

January 30, 2012



1. Question 1

Write a function $\text{binorm}(\mu_1, \mu_2, \sigma_1, \sigma_2, \text{corr})$ in R that returns a single pseudo-random vector \mathbf{X} from a bivariate Normal distribution with the specified means, standard deviations and correlation. You may use any excising R functions and libraries.

Solution :

In R we can sample from a multivariate normal distribution using the command $\text{rmvnorm}()$. To do that we first need to install the package *MASS*. This build-in function has arguments μ which is a vector for the mean of the bivariate normal distribution and Σ , the variance-covariance matrix of the bivariate normal distribution. Therefore since we are given the correlation coefficient as an input to the function, we need to compute the covariance of the components. Note that, for two random variables X_1 and X_2 , we have

$$\text{corr}(X_1, X_2) = \frac{\text{Cov}(X_1, X_2)}{\sigma_{X_1}\sigma_{X_2}}$$

$$\implies \text{Cov}(X_1, X_2) = \sigma_{X_1}\sigma_{X_2}\text{corr}(X_1, X_2)$$

```
binorm<-function(mu1,mu2,sigma1,sigma2,corr)
{#begin of function
  cov<-corr*sigma1*sigma2
  var.matrix<-matrix(c(sigma1^2,cov,cov,sigma2^2),ncol=2)
  mean.vec=c(mu1,mu2)
  x<-mvrnorm(n=1, mu=mean.vec ,Sigma=var.matrix,empirical = FALSE)
  return(x)
}#end of function
```

Using your binorm function, simulate a sample of size 100 from a bivariate Normal distribution with correlation -0.7 and means and standard deviations of your choice, and verify that your function appears to work correctly.

Solution :

To simulate a sample of 100 from the bivariate Normal distribution with correlation -0.7 we use the previous function.

```
binorm<-function(mu1,mu2,sigma1,sigma2,corr)
{#begin of function
  cov<-corr*sigma1*sigma2
  var.matrix<-matrix(c(sigma1^2,cov,cov,sigma2^2),ncol=2)
  mean.vec=c(mu1,mu2)
  x<-mvrnorm(n=100, mu=mean.vec,Sigma=var.matrix,empirical = FALSE)
  return(x)
}#end of function

>binorm(0,0,5,0.5,-0.7)
      [,1]      [,2]
[1,]  8.29976269  1.367240177
[2,]  3.11590473 -0.548470158

[omitted output]
```

We now need to check whether the function works correctly. Therefore we compute the observed means and variances of the Bivariate Normal components and then we plot the scatter plot and the Normal probability plots of the two components.

```
>x1<-x[,1] ; x2<-x[,2]
>mean(x1) ; mean(x2)
[1] -0.3255526
[1] -0.03099976
>var(x1) ; var(x2)
[1] 25.077
[1] 0.2543043
```

[For code of plots see appendix]

As we see from the above R output the observed means and variances are

$$\hat{\mu}_1 = -0.33 \text{ and } \hat{\mu}_2 = -0.03 \text{ which are very close to } \mu_1 = 0 \text{ and } \mu_2 = 0$$

$$\hat{\sigma}_1^2 = 25.1 \text{ and } \hat{\sigma}_2^2 = 0.254 \text{ which are very close to } \sigma_1^2 = 25 \text{ and } \sigma_2^2 = 0.25$$

To test for normality, we can check (*i*) the univariate marginals, by looking at the Normal probability plots and (*ii*) the bivariate marginals using two dimensional scatter plot. [Zygouras, page2] The scatter plot should resemble an ellipse and the points at the Q-Q plots should show a linear trend. As we see from the scatter plot in figure 1, the random points from the bivariate normal, have indeed the shape of an ellipse. Moreover the Q-Q plots of the components illustrated in figure 2, show a linear trend since the most of the points lie on the straight line, that passes through the 25th and 75th quantiles of the normal distribution. Hence we can conclude that the assumptions for bivariate normality don't significantly violated. Furthermore we can use the following property from the multivariate statistics theory, [Zygouras, page2] to further check the normality assumption.

Proposition 1.

The $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ assigns probability $(1 - \alpha)$ to the ellipsoid $\left\{ \mathbf{x} \in \mathbb{R}^p : (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq \chi_p^2(\alpha) \right\}$

This proposition can be applied using the following code.

```
> sum<-0
> for (j in 1:100)
+ {
+   x.vec<-c(x[j,1],x[j,2])
+   mean.vec<-c(mu1,mu2)
+   A<- (x.vec - mean.vec)
+   condition <- A %*% solve(var.matrix) %*% A
+   if (condition < qchisq(0.5,2) )
+     { sum<- sum + 1 }
+ }
> sum
[1] 53
```

As we see from the above output, we get 53 out of the 100 points that lie within the ellipse. Hence there aren't enough evidence to show violation of the normality assumption.

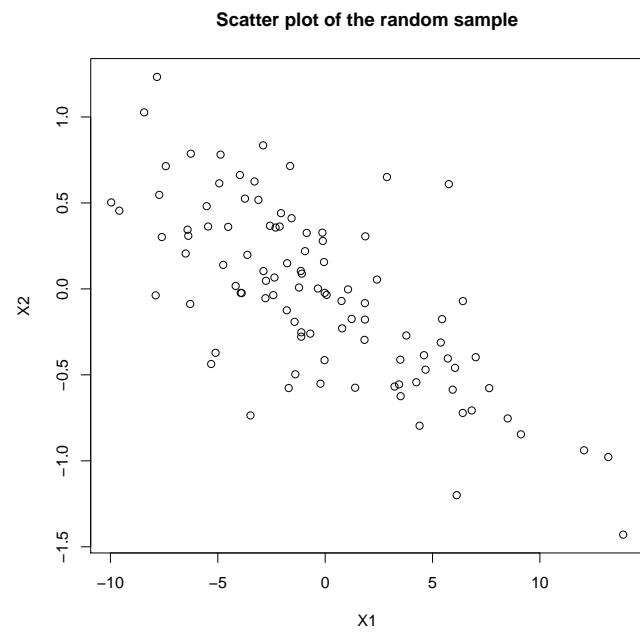


Figure 1: Scatter plot of the random sample

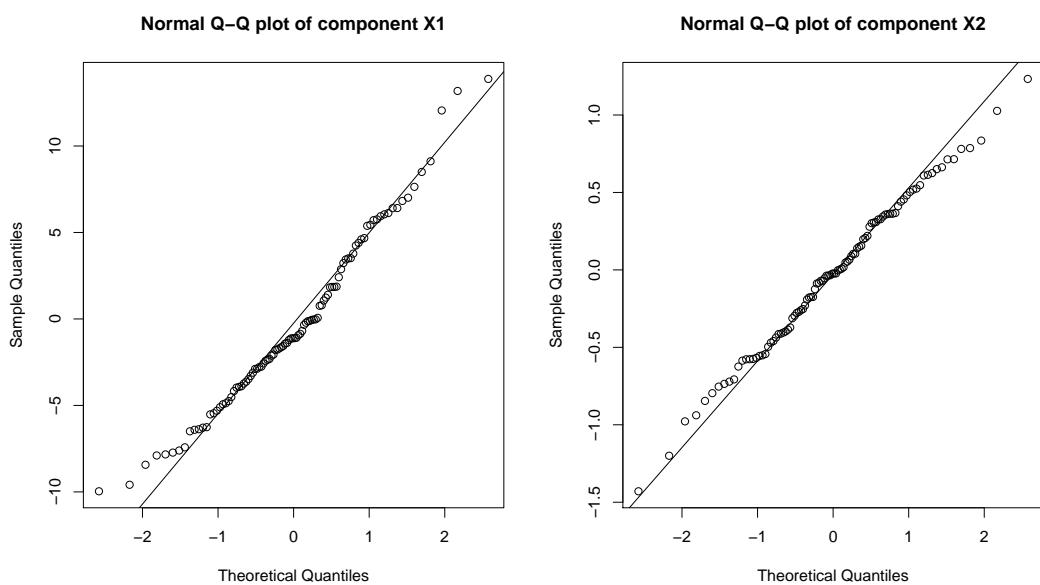


Figure 2: Normal probability plots

2. Question 2

Observations Y_i are assumed to come from Poisson distributions with corresponding means $m_i = \theta_1(1 - e^{-\theta_2 x_i})$, where both θ_1 and θ_2 are positive. Suppose the prior distribution for $\boldsymbol{\theta} = (\theta_1, \theta_2)^T$ has density

$$p(\boldsymbol{\theta}) \propto \frac{e^{-\frac{\theta_1}{10}}}{(1 + \theta_1^2 \theta_2^2)} \mathbb{1}\{\theta_1 > 0, \theta_2 > 0\}$$

The data are given in the following table:

Table 1: Data			
	i		
	1	2	3
x_i	1	3	10
y_i	13	28	41

Using random walk Metropolis-Hastings sampling with a bivariate Normal proposal, summarise the posterior distribution $p(\boldsymbol{\theta}|D)$. Also describe the process by which you arrived at your summary.

Solution :

We know that according to Bayes theorem,

$$\pi(\boldsymbol{\theta}|x) = \frac{f(x|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int f(x|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) d\boldsymbol{\theta}} \propto f(x|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$$

which means posterior \propto likelihood \times prior. Note that the posterior distribution in the Markov Chain Monte Carlo analysis is considered as the target distribution and the prior as the proposal distribution (or density). Therefore in this exercise we have,

$$p(\boldsymbol{\theta}|D) \propto L(\boldsymbol{\theta}|D) \times p(\boldsymbol{\theta})$$

Since Y_i are observations that come from Poisson distribution then the likelihood function is,

$$L(\boldsymbol{\theta}|D) = \prod_{i=1}^3 f(D|\boldsymbol{\theta}) = \prod_{i=1}^3 e^{-m_i} \frac{(m_i)^{y_i}}{y_i!} , \text{ where } m_i = \theta_1(1 - e^{-\theta_2 x_i})$$

$$\text{Note : } \prod_{i=1}^n \exp(x_i) = \exp(x_1) \dots \exp(x_n) = \exp(x_1 + \dots + x_n) = \exp\left(\sum_{i=1}^n x_i\right)$$

Hence, the likelihood becomes,

$$L(\boldsymbol{\theta}|D) = \frac{1}{\prod_{i=1}^3 y_i!} \exp\left\{-\sum_{i=1}^3 m_i\right\} \prod_{i=1}^3 m_i^{y_i}$$

$$\begin{aligned}
&= \frac{1}{y_1!y_2!y_3!} \exp \left\{ -\theta_1(1 - e^{-\theta_2 x_1}) - \theta_1(1 - e^{-\theta_2 x_2}) - \theta_1(1 - e^{-\theta_2 x_3}) \right\} \\
&\quad \times \left\{ [\theta_1(1 - e^{-\theta_2 x_1})]^{y_1} [\theta_1(1 - e^{-\theta_2 x_2})]^{y_2} [\theta_1(1 - e^{-\theta_2 x_3})]^{y_3} \right\} \\
&= \frac{1}{13!28!41!} \exp \left\{ -\theta_1(1 - e^{-\theta_2}) - \theta_1(1 - e^{-3\theta_2}) - \theta_1(1 - e^{-10\theta_2}) \right\} \\
&\quad \times \left\{ [\theta_1(1 - e^{-\theta_2})]^{13} [\theta_1(1 - e^{-3\theta_2})]^{28} [\theta_1(1 - e^{-10\theta_2})]^{41} \right\}
\end{aligned}$$

Using the proportionality rule (i.e ignore any constant terms) and rearranging we get,

$$\begin{aligned}
L(\boldsymbol{\theta}|D) &\propto \theta_1^{13+28+41} \exp \left\{ -\theta_1 + \theta_1 e^{-\theta_2} - \theta_1 + \theta_1 e^{-3\theta_2} - \theta_1 + \theta_1 e^{-10\theta_2} \right\} \\
&\quad \times \left\{ (1 - e^{-\theta_2})^{13} (1 - e^{-3\theta_2})^{28} (1 - e^{-10\theta_2})^{41} \right\} \\
&= \theta_1^{82} \exp \left\{ \theta_1(-3 + e^{-\theta_2} + e^{-3\theta_2} + e^{-10\theta_2}) \right\} \times \left\{ (1 - e^{-\theta_2})^{13} (1 - e^{-3\theta_2})^{28} (1 - e^{-10\theta_2})^{41} \right\}
\end{aligned}$$

Therefore, the posterior distribution is,

$$\begin{aligned}
p(\boldsymbol{\theta}|D) &\propto \theta_1^{82} \exp \left\{ \theta_1 \left(-\frac{31}{10} + e^{\theta_2} + e^{-3\theta_2} + e^{-10\theta_2} \right) \right\} \\
&\quad \times \left\{ (1 - e^{-\theta_2})^{13} (1 - e^{-3\theta_2})^{28} (1 - e^{-10\theta_2})^{41} \right\} \frac{1}{(1 + \theta_1^2 \theta_2^2)} \mathbb{1}\{\theta_1 > 0, \theta_2 > 0\}
\end{aligned}$$

Now the Metropolis-Hastings Algorithm as has been described in lectures is as follow. (This specific version is from Casella page 171) Note that $\pi(x)$ is the target density, $q(x^t; y)$ is the transition density, i.e the probability to fall in state y at time $t + 1$ given the chain is in state x at time t . This can be also written as $q(y|x^t)$ which shows more clearly the meaning of an event A given an event B. Also x_0 is the initial state of the chain and S is the state space.

Choose $X_0 = x_0 \in S$

Given $X^{(t)}$

1. Generate $Y_t \sim q(y|x^{(t)})$

2. Take

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability } \alpha(x^{(t)}, Y_t) \\ x^{(t)} & \text{with probability } 1 - \alpha(x^{(t)}, Y_t) \end{cases}$$

where

$$\alpha(x, y) = \min \left\{ \frac{\pi(y)q(x|y)}{\pi(x)q(y|x)}, 1 \right\}$$

The Random Walk Metropolis-Hastings sampler (*RWMH*) is defined by $q(x, y) = f(y - x)$. i.e $Y = X + Z$, where Z has density $f(\cdot)$.

The acceptance probability is

$$\alpha(x, y) = \min\left\{\frac{\pi(y)f(x-y)}{\pi(x)f(y-x)}, 1\right\}$$

Note that for a symmetric proposal, like the case of the bivariate normal distribution $f(y - x) = f(x - y)$. Hence we can simplify the acceptance probability by not including that term. Also since $Z \sim \mathcal{N}_2(\mu, \Sigma)$ then $Y = (X + Z) \sim \mathcal{N}_2(x + \mu, \Sigma)$. Therefore, using the Random Walk Metropolis-Hastings sampler, we can investigate for adequate parameters of the proposal distribution, so that the chain converge to the stationary distribution, which is also the target (or the posterior) distribution that we want to sample from. The code is as follow.

```
#function to sample from the Bivariate Normal distribution
binorm <- function( mu1, mu2, sigma1, sigma2, corr)
{
  #begin of function
  cov<- corr*sigma1*sigma2
  var.matrix<-matrix(c(sigma1^2,cov,cov,sigma2^2), ncol=2)
  mean.vec=c(mu1,mu2)
  x<-mvrnorm(n=1, mu=mean.vec, Sigma_matrix,empirical=FALSE)
  return(x)
}#end of function

#function to get the density of the posterior distribution
posterior<- function(x)
{
  #begin of function
  if (x[1,1]>0 && x[1,2]>0) #condition of the indicator function
  {
    U1 <- (x[1,1]^82) / (1 + (x[1,1]^2)*(x[1,2]^2))
    U2 <- exp( x[1,1]*(-31/10 + exp(-x[1,2]) + exp(-3*x[1,2]) + exp(-10*x[1,2])) )
    U3 <- (1 - exp(-x[1,2])) ^13
    U4 <- (1 - exp(-3*x[1,2])) ^28
    U5 <- (1 - exp(-10*x[1,2])) ^41
    posterior.density <- U1*U2*U3*U4*U5
  }
  else posterior.density <- 0
  return(posterior.density)
}#end of function

#function to sample using RWMH sampler
RWMH <- function(N, mu1, mu2, sigma1, sigma2, corr, init.1, init.2)
{
  #begin of function
  x<- matrix(0, N, 2)
  x[1,1]<- init.1 #keeps the updates for theta1
  x[1,2]<- init2 #keeps the updates for theta1
  count<-0
  for (i in 2:N)
  {
    #begin of for loop
    y<- binorm( mu1 + x[i-1,1], mu2 + x[i-1,2], sigma1, sigma2, corr)
    alpha.numer <- posterior( t(y) )
    #end of for loop
    if (runif(1)<alpha.numer)
      x[i,] <- y
    else
      x[i,] <- x[i-1,]
  }
  return(x)
}#end of function
```

```

alpha.denom <- posterior( t(as.matrix(x[i-1, ])) )
alpha <- alpha.numer / alpha.denom
alpha <- min(1,alpha)
u <- runif(1)
if (u < alpha) #the acceptance probability condition
{
  x[i, ] <- y
  count <- (count + 1) #count for the no. of acceptance densities
}
else x[i, ] <- x[i-1, ]
}#end of for loop
print(count/N) #the acceptance ratio
return(x)
}#end of function

>chain1 <- RWMH(100000, 0, 0, 10, 0.5, -0.7, 50, 0.25)

>theta.1 <- chain1[ , 1] ; theta.2 <- chain1[ , 2]
>par(mfrow=c(2,2))
>ts.plot(theta.1) ; ts.plot(theta.2)
>acf(theta.1) ; acf(theta.2)

```

The diagnostic plots are useful to check whether the chain has sufficiently explored the posterior distribution and the sequence of sampling points have converged. [Albert page 107] Another way for someone to understand how the *MCMC* convergence works is to think this like the mixing of two chemicals (the 2 populations) until they become stable and then give a new stable product. (Turing stability) Therefore, by looking at the trace plots we can see if the true posterior distribution has been approached.

Furthermore the autocorrelation of the chain is another measure that can be considered when we want to check its convergence. The autocorrelation function is given by

$$\gamma(k) = \text{Cov}(h(X_t), h(X_{t+k})), \quad k \in \mathbb{Z}$$

and the autocorrelation function is

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)}, \quad k \in \mathbb{Z}$$

Note that for $k = 0$ we have $\gamma(0) = \text{Cor}(h(X_t), h(X_t)) = \text{Var}(h(X_t))$ and $\rho(k) = \frac{\gamma(0)}{\gamma(0)} = 1$.

Autocorrelation is a measure of the degree of dependency between successive values in the chain, i.e for example X_t and X_{t+k} where k the lag or the number of iterations separating the two values. If the chain is mixing well, then the values of the autocorrelation will decrease to zero as the lag value is increased. (i.e the x-axis on the plots) On the other hand strong correlation might show that the chain doesn't explore the entire parameter space.

Choosing the parameters of the proposal distribution, is based on trial and error. After several trials a summary of the main points for consideration is presented here. An example of chain that hasn't converged is illustrated in figure 3. This chain had small variances, which means that the Markov Chain will be highly correlated. Similarly if we choose large variances then the Markov Chain will move very fast. Ideally we want variances not too large but not too small either. We want to avoid any acceptance ratio that is very small, like for example 0.03 or very large like 0.75 that we have observed

in same cases. As we see from the trace plots of θ_1 and θ_2 in figure 3, the plots show some smooth trend, especially the trace plot of θ_1 . This is an evidence that the chain doesn't converge. The plot should be oscillating very fast and no trend should be shown.

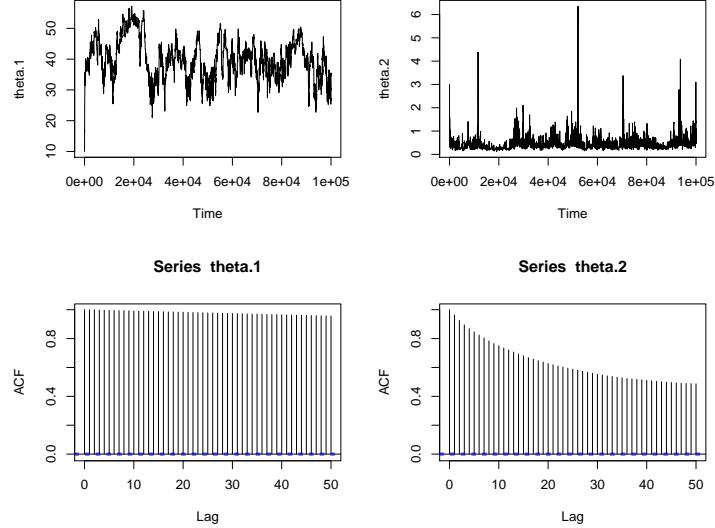


Figure 3: An example of a chain that hasn't converged

Some examples of different values that have been tried out are shown on table 2.

Table 2: Trial values for the parameters

N	μ_1	μ_2	σ_1	σ_2	ρ	θ_1	θ_2	α
10^5	0	0	10	2	-0.7	100	4	0.06
10^5	0	0	5	0.5	-0.7	38	0.8	0.25
10^5	0	0	10	0.5	-0.7	39	0.7	0.22
10^5	0	0	10	0.25	-0.7	40	0.7	0.343
10^5	0	0	10	0.25	-0.7	39	0.6	0.346
10^5	0	0	10	0.25	-0.7	37	0.6	0.343

Therefore by choosing the parameters $\mu_1 = 0$, $\mu_2 = 0$, $\sigma_1 = 10$, $\sigma_2 = 0.25$, $\rho = -0.7$, $\theta_1 = 38$ and $\theta_2 = 0.7$ we can plot the trace plots and autocorrelation plots of the chain. This can be seen on figure 4. As we can observe, the trace plots oscillate very fast without showing any trend and the autocorrelation functions decreases as the lag increases. These are good evidence for convergence. Moreover since, these are necessary but not sufficient conditions, we can also perform convergence diagnostics. This can be done in R by using the `boa.menu()`. To analyse the chain and check for convergence to the stationary distribution we use 2 methods, the Geweke method and the Raftery and Lewis method. Their output are shown below.

GEWEKE CONVERGENCE DIAGNOSTIC:

=====

```
Fraction in first window = 0.1
Fraction in last window = 0.5
```

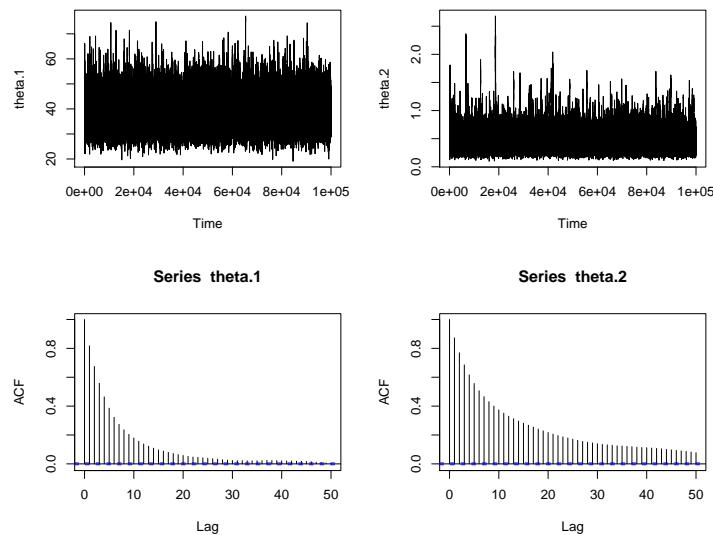


Figure 4: An example of a chain that has converged

Chain: chain1

```
par1      par2
Z-Score -1.1808553 1.1607167
p-value  0.2376602 0.2457571
```

RAFTERY AND LEWIS CONVERGENCE DIAGNOSTIC:

```
Quantile = 0.025
Accuracy = +/- 0.005
Probability = 0.95
```

Chain: chain1

	Thin	Burn-in	Total	Lower Bound	Dependence Factor
par1	9	27	34704	3746	9.264282
par2	13	39	49543	3746	13.225574

The Geweke's method consider 2 sections of the chain. That is, it consider a section at the beginning of the chain after burn-in and a section at the end of the chain. If the chain has converged then the 2 means should be similar and sequences of individual terms should have similar properties. [Shaw page 36] From the output of Geweke's diagnostic we see the z-scores and the p-values for the parameters θ_1 and θ_2 . This method test the null hypothesis that the 2 averages are equal versus the alternative hypothesis the 2 averages are different. We can reject the null hypothesis at significance level $\alpha = 0.05$ if p-value ≤ 0.05 . Since for both cases the p-value is greater than 0.05, there are not

enough evidence to reject the null hypothesis and therefore we conclude that the means are the same. This shows that convergence of the Markov Chain have been achieved.

Additionally, the Raftery and Lewis method has the parameters M = burn-in length, N = total number of iterations and K="thinning factor". (which means that the only every k^{th} iteration is used for estimation). As we see from the Raftery and Lewis convergence diagnostic output, the burn-in value for θ_1 is 27 and the burn-in value for θ_2 is 39. These are very small numbers and this may be due to fact that the method is looking at a lower quantile of the target distribution. Also it suggested a high number of iterations until the reach of burn-in period, due to the very large dependence factor. After all, this method might not be appropriate for detecting the problem caused by sticky modes.

Summarising all the above analysis, the chosen proposal distribution is,

$$\mathcal{N}_2 \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 100 & -0.7 \\ -0.7 & 100 \end{pmatrix} \right)$$

and the chosen parameters for the posterior distribution with pdf as given before are $\theta_1 = 37$ and $\theta_2 = 0.7$. In figure 5, we also see the scatter plot of the 2 parameters. Note that it has an exponential form as expected because of the form of the posterior distribution pdf. Furthermore we can compute the mean of the posterior distribution using the ergodic theorem. The theorem says that if we have a random variable $h(Y)$ where Y has the a distribution given by density π then we can estimate $E(h(Y))$ by $\frac{1}{N} \sum_{n=1}^N h(X_n)$. In other words for sufficiently large N, our estimate will be close to $E(h(Y))$.

To compute the first two moments of the posterior distribution note that the moments for a random variable X are given by,

$$\mathbb{E}[X^r] \text{ where } r = 1, 2, \dots$$

Hence for $r=1$ we get the first moment which is just the expectation of the random variable. The second moment is $\mathbb{E}[X^2]$ and therefore the variance can be computed with $Var[X] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$. The approximated mean for the parameters of the posterior distribution after the burn-in period let say 40, is computed using the following code.

```
> summation1<-0
> for (k in 40:100000)
+ {
+   summation1<- summation1 + theta.1[k]
+ }
> ave.theta1<- summation/(100000-40)
> ave.theta1
[1] 77.25995
>
> summation2<-0
> for (k in 40:100000)
+ {
+   summation2<- summation2 + theta.2[k]
+ }
> ave.theta2<- summation2/(100000-40)
> ave.theta2
[1] 0.4515701
```

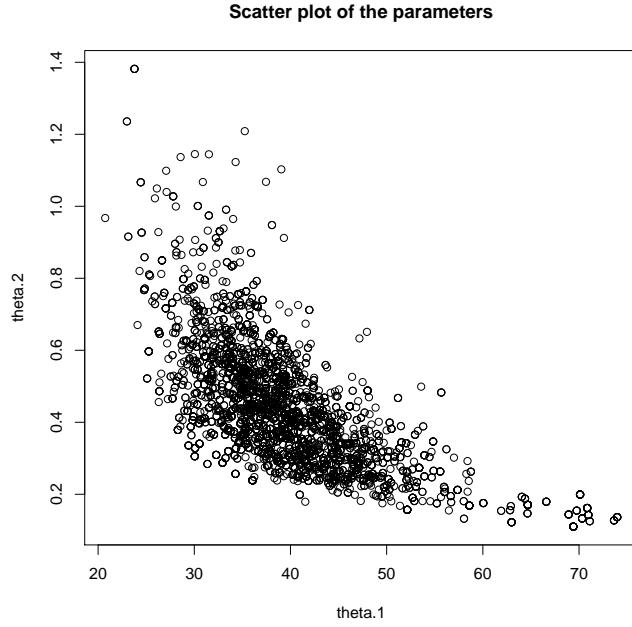


Figure 5: Scatter plot of θ_1 Vs θ_2

3. Question 3

Investigate (briefly) the effect of changing

(a) The prior. For example, what happens with a vague prior $p(\theta) \propto \frac{1}{(1+\theta_1^2\theta_2^2)} \mathbb{1}\{\theta_1 > 0, \theta_2 > 0\}$

Solution :

In this exercise we want to check the effect of changing the prior. For example we choose a prior as in exercise 2, but without the term $e^{-\frac{\theta_1}{10}}$. Therefore we use the same code as before with only difference the new prior. The part of the code that has been changed is shown below.

```
#function to get the density of the posterior distribution
posterior<- function(x)
{
  #begin of function
  if (x[1,1]>0 && x[1,2]>0) #condition of the indicator function
  {
    U1 <- (x[1,1]^82) / (1 + (x[1,1]^2)*(x[1,2]^2))
    U2 <- exp( x[1,1]*(-3 + exp(-x[1,2]) + exp(-3*x[1,2]) + exp(-10*x[1,2])) )
    U3 <- (1 - exp(-x[1,2])) ^13
    U4 <- (1 - exp(-3*x[1,2])) ^28
    U5 <- (1 - exp(-10*x[1,2])) ^41
    posterior.density <- U1*U2*U3*U4*U5
  }
  else posterior.density <- 0
  return(posterior.density)
}#end of function
```

We then try similar values for the parameters as we have tried before, and we observe that the accep-

tance ratio is similar with the one we got before. For example for $\mu_1 = 0$, $\mu_2 = 0$, $\sigma_1 = 10$, $\sigma_2 = 0.25$, $\rho = -0.7$, $\theta_1 = 40$, $\theta_2 = 0.9$ we get $\alpha = 0.312$. The trace and autocorrelation plots for parameters θ_1 and θ_2 , as they are shown in figure 6, give visual evidence that the chain has converged. In other words, by changing the prior we don't get dramatically different results. However some extreme values are observed. Changing the prior might not always give the same or better results.

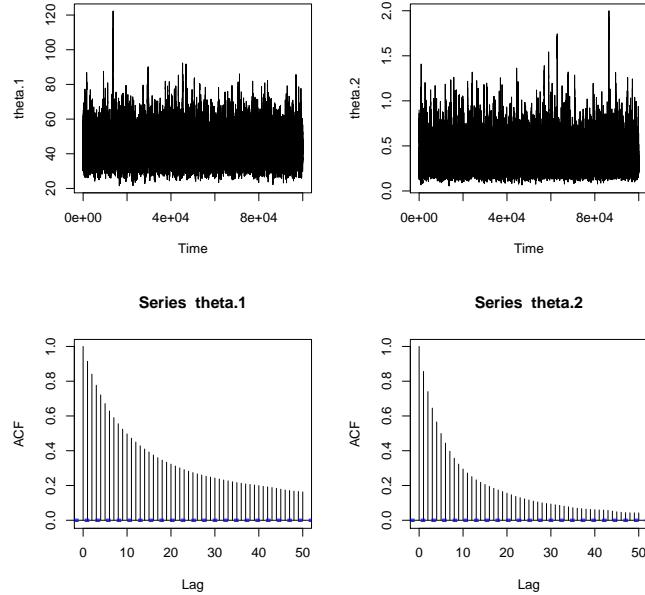


Figure 6: Trace plots and autocorrelation plots for the new prior

(b) The MH Kernel.

Solution :

If $Z = (Z_1, Z_2)^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $U \& Z$ are independent, with $U = R$ or $U = \frac{1}{R}$ where $R \sim \text{Gamma}(\mu = 1, \theta)$, $\mu > 0$ the mean and $\theta > 0$ the shape parameter, then UZ has circularly symmetric density, and may be transformed to have an elliptically symmetric density.

In other words, U will be a random number generated from a gamma distribution with mean equals to 1. Therefore Z , which has a bivariate standard normal distribution, multiplied by this random number will give a linear transformation of the circularly symmetric density of Z to an elliptically symmetric density.

Note that the pdf of a random variable X that follows a gamma distribution is,

$$f(x; \alpha, \beta) = \frac{1}{\Gamma(\alpha)} \frac{x^{\alpha-1}}{\beta^\alpha} \exp\left\{-\frac{x}{\beta}\right\}$$

where $\alpha > 0$ is the shape parameter and $\beta > 0$ is the scale parameter.

Also, $\mathbb{E}(X) = \alpha\beta$. Now since we are given that the mean equals to 1 then it means that $\alpha = \frac{1}{\beta}$. Therefore the parameters of the gamma should be chosen such that they are inverse, i.e for example $\alpha = 3$ and $\beta = \frac{1}{3}$.

We know from linear algebra theory that in order to apply a linear transformation to a vector, we need to use the rotation matrix. This is given by

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

Recall that for the Random Walk Metropolis Hastings sampler the kernel is $\mathbf{Y} = \mathbf{X} + \mathbf{Z}$. Thus, in this example becomes $\mathbf{Y} = \mathbf{X} + U\mathbf{Z}$.

Hence, $\mathbb{E}[\mathbf{Y}] = \mathbf{X} + U\mathbb{E}[\mathbf{Z}]$ where $\mathbb{E}[\mathbf{Z}] = (\mu_1, \mu_2)^T$ and $Var[\mathbf{Y}] = U^2Var(\mathbf{Z})$

The code for the *RWMH* with the new kernel is as follow.

```
#function to sample using RWMH sampler
RWMH <- function(N, mu1, mu2, sigma1, sigma2, corr, init.1, init.2, theta, shape.gam)
  {#begin of function
    x<- matrix(0, N, 2)
    x[1,1]<- init.1 #keeps the updates for theta1
    x[1,2]<- init2 #keeps the updates for theta1
    count<-0
    rotation<-matrix(c(cos(theta),-sin(theta),sin(theta),cos(theta)),nrow=2)
    for (i in 2:N)
      {#begin of for loop
        U<-rgamma(n=1,shape.gam,(1/shape.gam))
        y<- binorm( U*mu1 + x[i-1,1], U*mu2 + x[i-1,2], U*sigma1, U*sigma2, (U^2)*corr)* rotation
        alpha.numer <- posterior( t(y) )
        alpha.denom <- posterior( t(as.matrix(x[i-1, ])) )
        alpha <- alpha.numer / alpha.denom
        alpha <- min(1,alpha)
        u <- runif(1)
        if (u < alpha) #the acceptance probability condition
        {
          x[i, ]<- y
          count <- (count + 1) #count for the no. of acceptance densities
        }
        else x[i, ]<- x[i-1, ]
      }#end of for loop
    print(count/N) #the acceptance ratio
    return(x)
  }#end of function

>Chain2 <- RWMH(5000, 0, 0, 1, 1, 1, 40, 1, pi/4, 3 )
```

References

- [1] Jim Albert.(2007) *Bayesian Computation with R*. Springer.
- [2] Christian P.Robert and George Casella.(2010) *Introducing Monte Carlo Methods with R*. Springer.
- [3] Elke Thonnes and Ewart Shaw. (2011) *Monte Carlo Methods*. Lectures notes of ST407. University of Warwick,UK, unpublished.
- [4] Nikos Zygouras. (2011) *Multivariate Statistics*. Lectures notes of ST412. University of Warwick,UK, unpublished.
- [5] R Development Core Team. *R: A Language and Environment for Statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL <http://www.r-project.org/>.

Appendices

A. Question 1 - R code for plots

```
#The scatter plot of the random sample  
>plot(x,xlab="X1",ylab="X2",main="Scatter plot of the random sample" )  
  
>par(mfrow=c(1,2))  
#The normal Q-plots for each component  
>qqnorm(x1,main="Normal Q-Q plot of component X1")  
>qqline(x1)  
  
>qqnorm(x2,main="Normal Q-Q plot of component X2")  
>qqline(x2)
```