David Giles

Bayesian Econometrics

6. More Bayesian Computation

Non-Standard Example

- Our data come from a Poisson process with an *unknown* break-point.
- $Y_i \sim Poi.(\theta_1)$; $i = 1, 2, 3, \ldots\ldots., m$; $\theta_1 > 0$
- $Y_i \sim Poi.(\theta_2)$; $i = m+1, 2, 3, \ldots\ldots.,n$; $\theta_2 > 0$
- θ_1 ; θ_2 ; and m are the unknown parameters.

- Prior distribution:

$$p(\theta_1, \theta_2, m) = p(\theta_1)p(\theta_2)p(m)$$

- $p(\theta_1)$ and $p(\theta_2)$ are Gamma; $p(m)$ is discrete Uniform.
- $p(\theta_1) \propto {\theta_1}^{a_1-1} e^{-\theta_1/b_1}$; a_1 , $b_1 > 0$; shape & scale
- $p(\theta_2) \propto {\theta_2}^{a_2-1} e^{-\theta_2/b_2}$; a_2 , $b_2 > 0$; shape & scale
- $E[\theta_i] = a_i b_i$; $Var.[\theta_i] = a_i {b_i}^2$; $i = 1, 2.$
- $p(m) = 1/n$; $m = 1, 2, \ldots\ldots, n$
- So, the joint prior for the 3 parameters is

$$p(\theta_1, \theta_2, m) \propto {\theta_1}^{a_1-1} e^{-\theta_1/b_1} {\theta_2}^{a_2-1} e^{-\theta_2/b_2}$$

- The Likelihood function is

$$L(\theta_1, \theta_2, m \,|\boldsymbol{y}) = p(\boldsymbol{y} \mid \theta_1, \theta_2, m)$$

$$= \prod_{i=1}^{m} [{\theta_1}^{y_i} e^{-\theta_1} / y_i!] \prod_{i=m+1}^{n} [{\theta_2}^{y_i} e^{-\theta_2} / y_i!]$$

The Likelihood function can be written as:

$$L(\theta_1, \theta_2, m \,|\boldsymbol{y}) = e^{-m\theta_1 - (n-m)\theta_2} \prod_{i=1}^{m} [{\theta_1}^{y_i} / y_i!] \prod_{i=m+1}^{n} [{\theta_2}^{y_i} / y_i!]$$

$$\propto e^{-m\theta_1 - (n-m)\theta_2} \left({\theta_1}^{\sum_1^m (y_i)}\right) \left({\theta_2}^{\sum_{m+1}^n (y_i)}\right)$$

- Apply Bayes' Theorem:

$$p(\theta_1, \theta_2, m \,|\boldsymbol{y}) \propto p(\theta_1, \theta_2, m) L(\theta_1, \theta_2, m \,|\boldsymbol{y}) \qquad \propto$$

$$\theta_1{}^{a_1-1} e^{-\theta_1/b_1} \theta_2{}^{a_2-1} e^{-\theta_2/b_2} e^{-m\theta_1-(n-m)\theta_2} \left(\theta_1{}^{\sum_1^m (y_i)}\right) \left(\theta_2{}^{\sum_{m+1}^n (y_i)}\right)$$

- Consider the *conditional posteriors* for the three parameters:
- Two of these are *densities*, and one is a *mass function*.
- $p(\theta_1 \mid \theta_2, m\,, \boldsymbol{y}) \propto \theta_1{}^{a_1-1} e^{-\theta_1/b_1} e^{-m\theta_1} \left(\theta_1{}^{\sum_1^m (y_i)}\right)$

$$\propto \theta_1{}^{a_1+\sum_1^m (y_i)-1} e^{-\theta_1(m+\frac{1}{b_1})}$$

This is the kernel of a Gamma density:

Shape parameter = $a_1 + \sum_1^m (y_i)$

Scale parameter = $(m + \frac{1}{b_1})^{-1}$

Note that $p(\theta_1 \mid \theta_2, m, \boldsymbol{y}) = p(\theta_1 \mid m, \boldsymbol{y})$

- $p(\theta_2 \mid \theta_1, m, \boldsymbol{y}) \propto {\theta_2}^{a_2-1} e^{-\theta_2/b_2} e^{-(n-m)\theta_2} \left({\theta_2}^{\sum_{m+1}^{n}(y_i)}\right)$

$$\propto {\theta_2}^{a_2+\sum_{m+1}^{n}(y_i)-1} e^{-\theta_2((n-m)+\frac{1}{b_2})}$$

This is also the kernel of a Gamma density:

Shape parameter = $a_2 + \sum_{m+1}^{n}(y_i)$

Scale parameter = $((n-m)+\frac{1}{b_2})^{-1}$

Note that $p(\theta_2 \mid \theta_1, m, \boldsymbol{y}) = p(\theta_2 \mid m, \boldsymbol{y})$

- The conditional posterior mass function for *m* is totally non-standard:

$$p(m \mid \theta_1, \theta_2, \boldsymbol{y}) \propto e^{-m\theta_1-(n-m)\theta_2}\left(\theta_1^{\sum_1^m (y_i)}\right)\left(\theta_2^{\sum_{m+1}^n (y_i)}\right)$$

$$\propto e^{-m(\theta_1-\theta_2)}\left(\theta_1^{\sum_1^m (y_i)}\right)\left(\theta_2^{\sum_1^n (y_i)}\right)\left(\theta_2^{-\sum_1^m (y_i)}\right)$$

$$\propto e^{-m(\theta_1-\theta_2)}(\theta_1/\theta_2)^{\sum_1^m (y_i)}$$

- How are we going to generate random draws from this mass function?

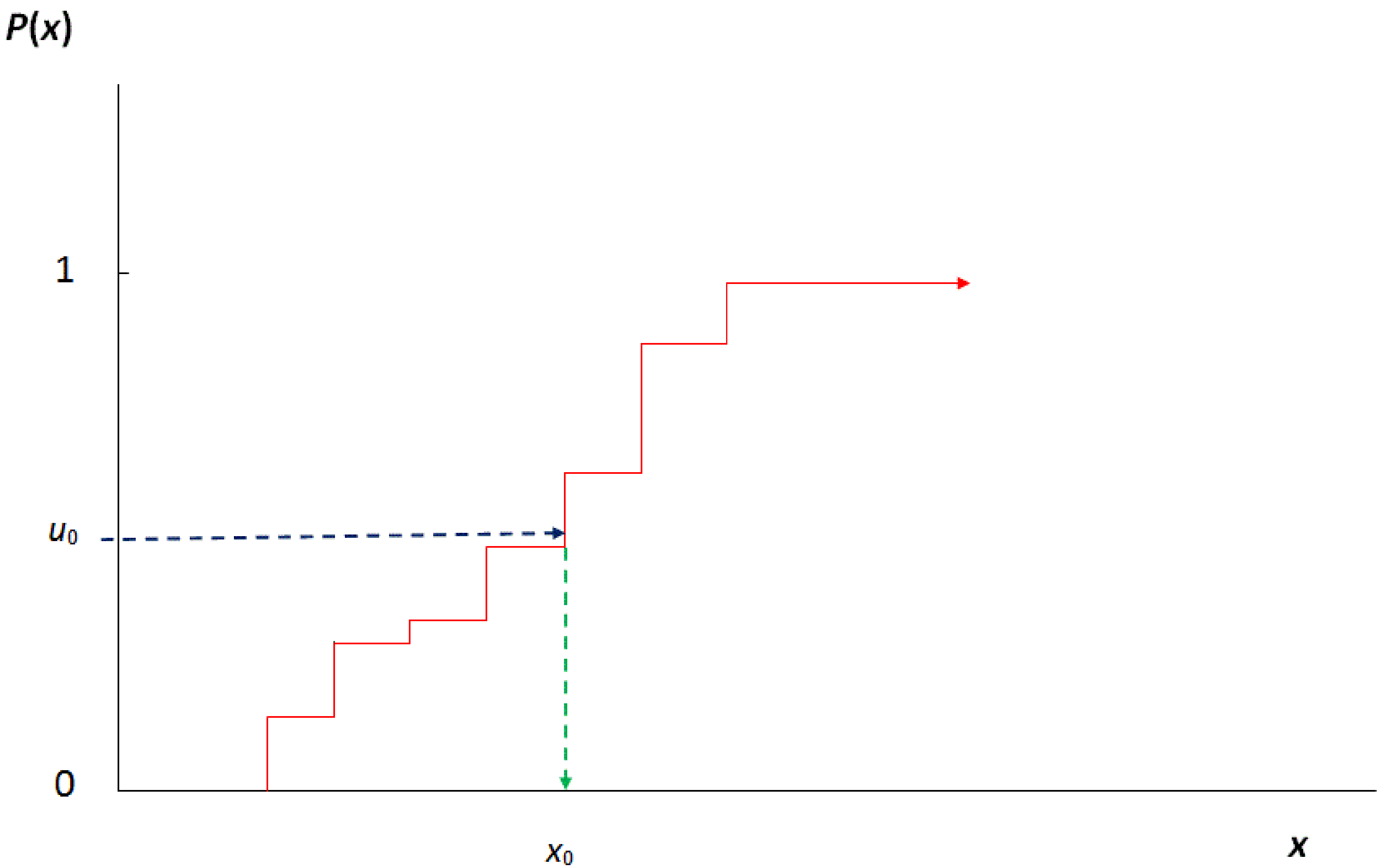
Table Look-up Method

- Useful for generating either continuous or discrete random variables from a *non-standard* distribution.
- Illustrate with discrete case, but continuous case is essentially the same.
- Essentially a numerical approach to "inverting" the distribution function - first suggested von Neumann.
- Suppose X is discrete on $[1 , n]$, say.

(i) Create c.d.f. for X, $P(x)$. This will be a non-decreasing function of x, with a lower limit of 0 and an upper limit of 1.

(ii) Generate a random number, say u_0, from $U\,[0 , 1]$.

(iii) Let x_0 be the value of x such that $x_0 = \max.\{x\colon u_0 \leq P(x)\}$.



P(x)
1
u0
0
x0
x

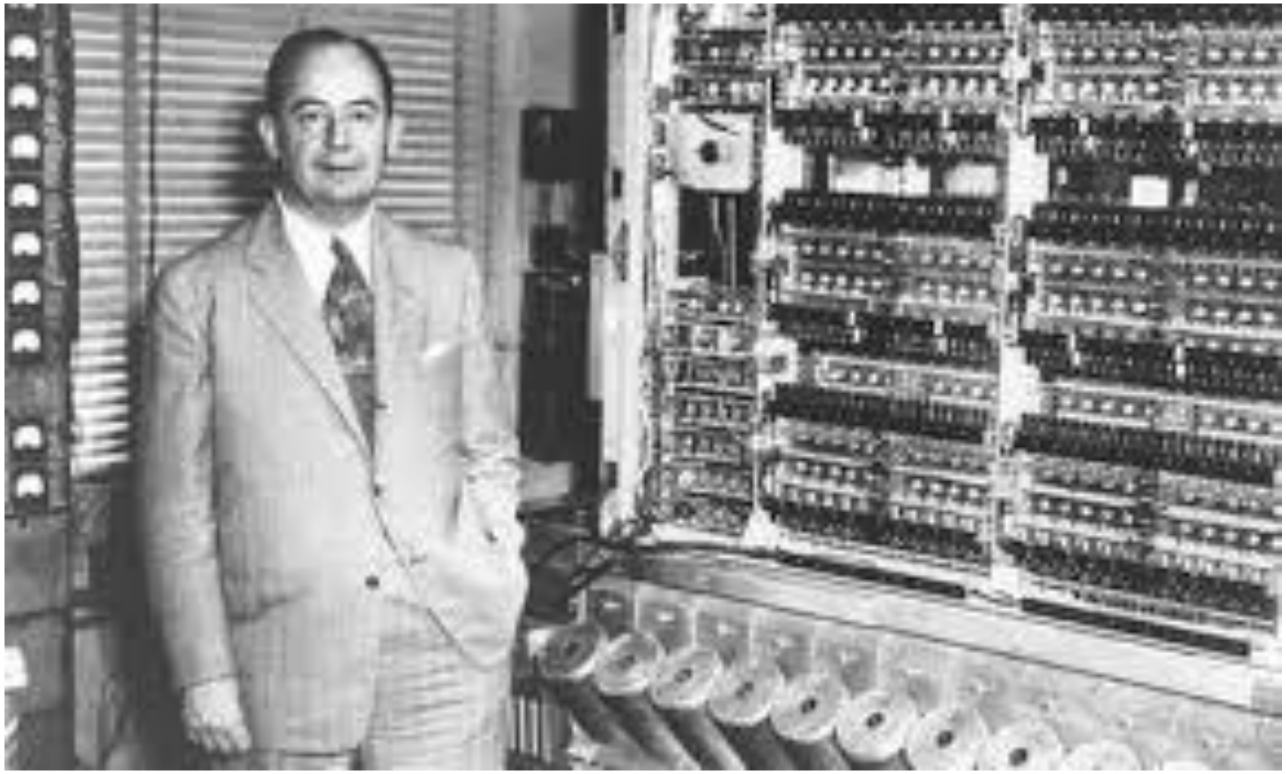
***x*:**	1	2	3	4	5	....	n
***P*(*x*):**	0.14	0.25	0.41	0.72	0.86	....	1.00

If $u_0 = 0.32$,

then $x_0 = 3$

If $u_1 = 0.81$,

then $x_1 = 5$; *etc*.



John von Neumann (1903 - 1957)

"Anyone who attempts to generate random numbers by deterministic means is, of course, living in a state of sin."

- Apply the Look-up table method to our problem, to generate values from the non-standard conditional posterior p.m.f. for the break-point, "*m*".
- The steps needed (for given values of the other 2 parameters), are:

(i) Calculate the values of the kernel of the conditional posterior,

$$k(m) = e^{-m(\theta_1-\theta_2)}(\theta_1/\theta_2\,)^{\sum_1^m\,(y_i)} \quad ; \quad m = 1, 2, \ldots\ldots\ldots, n$$

(ii) *Normalize* these values to get the conditional posterior p.m.f. itself:

$$p(m) = k(m)/\sum_{j=1}^{n} k(j) \quad ; \quad m = 1, 2, \ldots\ldots\ldots, n$$

(iii) Calculate the *cumulative* conditional posterior mass function for "*m*":

$$P(m) = \sum_{j=1}^{m} p(j) \quad ; \quad m = 1, 2, \ldots\ldots\ldots, n$$

- Now we have the Look-up Table and we can use it for generating random values of *m*.

R Code for Constructing Look-up Table

```
# Define vectors for storing results:
mm<- vector(length=n)
pm<- vector(length=n)
cusum<- vector(length=n)
# Construct the Look-up Table itself
for (i in 1:n)  {
  mm[i] <- exp(-i*(theta1-theta2))*(theta1/theta2)^(sum(y[1:i]))
}
pm <- mm / sum(mm)
cusum <- cumsum(pm)
# Now we have the look-up table for "m" !
```

Full R Code for Gibbs Sampler

```
library(moments)
library(modeest)
set.seed(1235)
nrep<- 50000
burnin<- 2000
n1<- 3          # Create artificial sample of data
n2<- 7
n<- n1+n2
y1<- rpois(n1,4)
y2<- rpois(n2,1)
y<-c(y1,y2)
```

```
#  Set initial values for Theta1 and Theta2
theta1<-  theta2<- 1
# Assign parameters for the priors for Theta1 and Theta2
a1<- 50       # shape parameters
a2<- 10
b1<- 0.1       # scale parameters
b2<- 0.1
margm<- vector(length=nrep)   # Set up vectors for later use
margt1<- vector(length=nrep)
margt2<- vector(length=nrep)
mm<- vector(length=n)
pm<- vector(length=n)
cusum<- vector(length=n)
```

```
# Now start the Gibbs Sampler:

for (ii in 1:nrep)  {

# Create the "Look-up Table" for the conditional p.m.f. of "m"
for (i in 1:n)  {
mm[i]<- exp(-i*(theta1-theta2))*(theta1/theta2)^(sum(y[1:i]))
 }
pm<-mm/sum(mm)
cusum<- cumsum(pm)
#  Now we have the look-up table
```

```
#  Generate a value of "m":
u<- runif(1, 0,1)
margm[ii]<- m<- which.max(u<cusum)
# Generate a value for "Theta1":
sy1<- sum(y[1:m])
margt1[ii]<- theta1<- rgamma(1,shape=(a1+sy1),
scale=1/(m+(1/b1)))
# Generate a value for "Theta2" (Allow for possibility that m=n)
sy2<- sum(y[min((m+1),n):n])
margt2[ii]<-theta2<- rgamma(1,shape=(a2+sy2), scale=1/((n-
m)+(1/b2)))

}           # End of the Gibbs Sampler
```

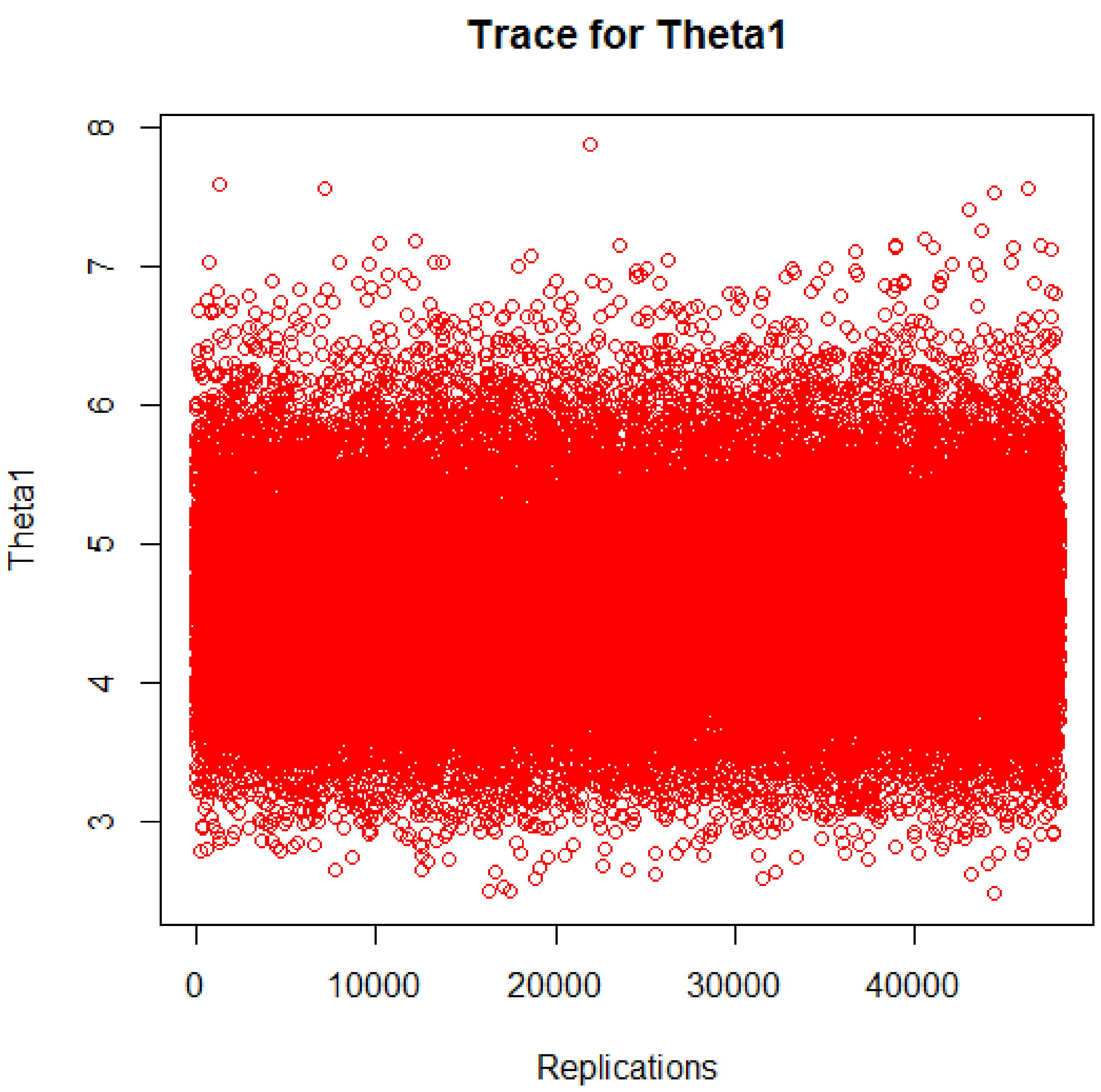
Some Illustrative Results

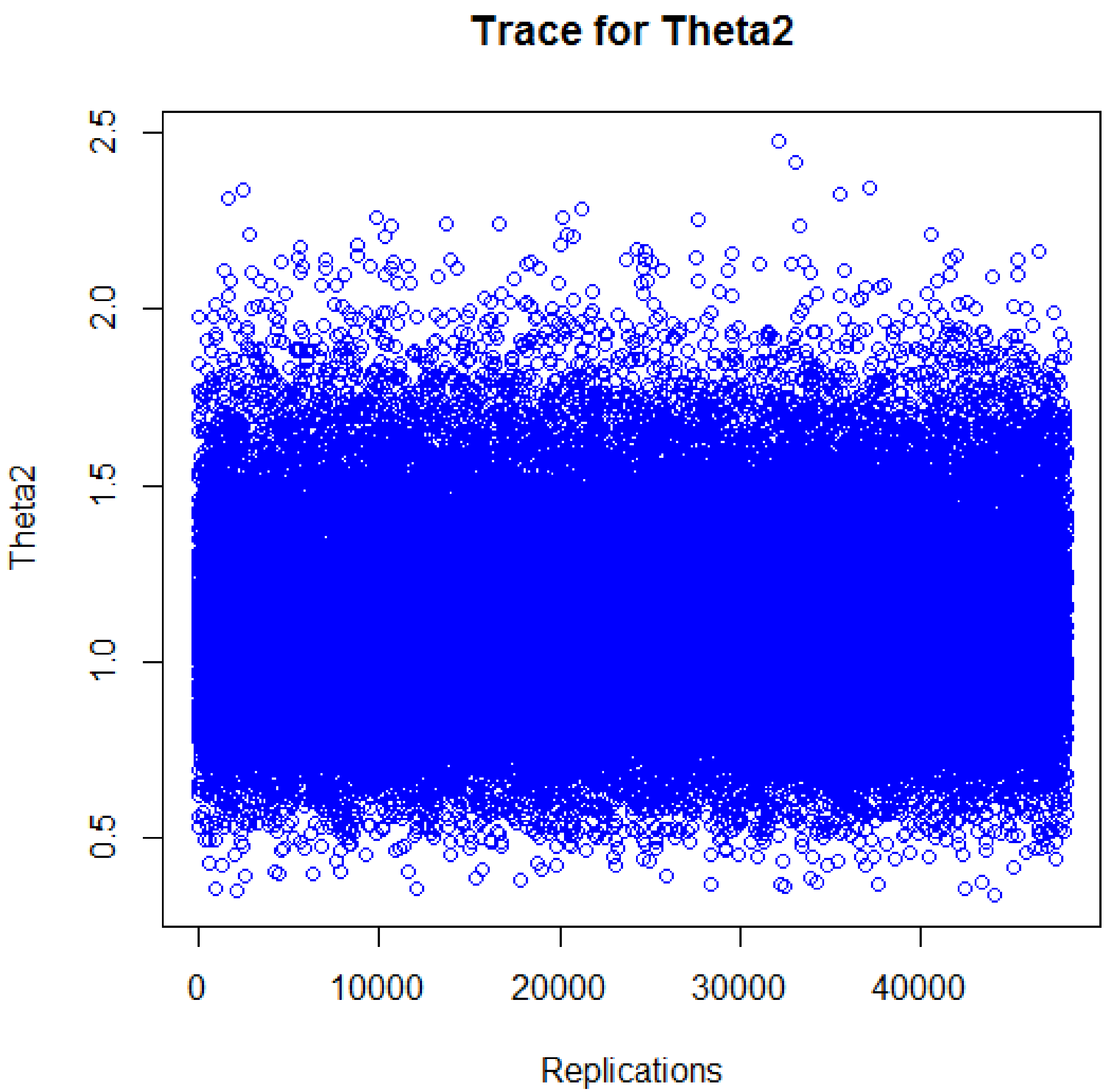
- Data-generating process: $n_1 = 3$; $n_2 = 7$; so, $m = 4$
- Gibbs sampler - 50,000 draws; Burn-in = 2,000 draws
- Prior parameters:

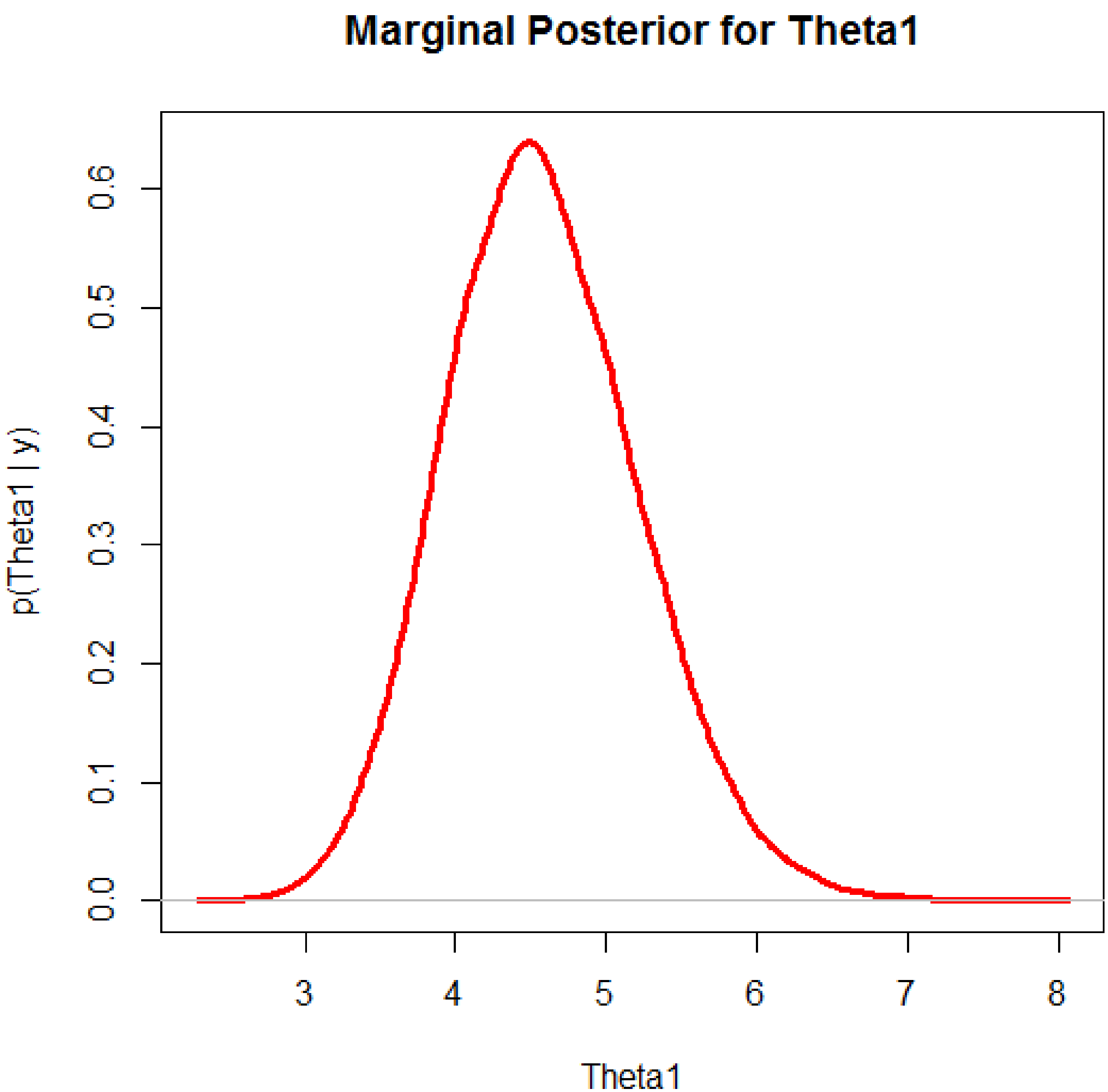
$a1 = 50$; $b1 = 0.1$ ⇒ $E(\theta_1) = 5\,; \; Var.(\theta_1) = 0.5$

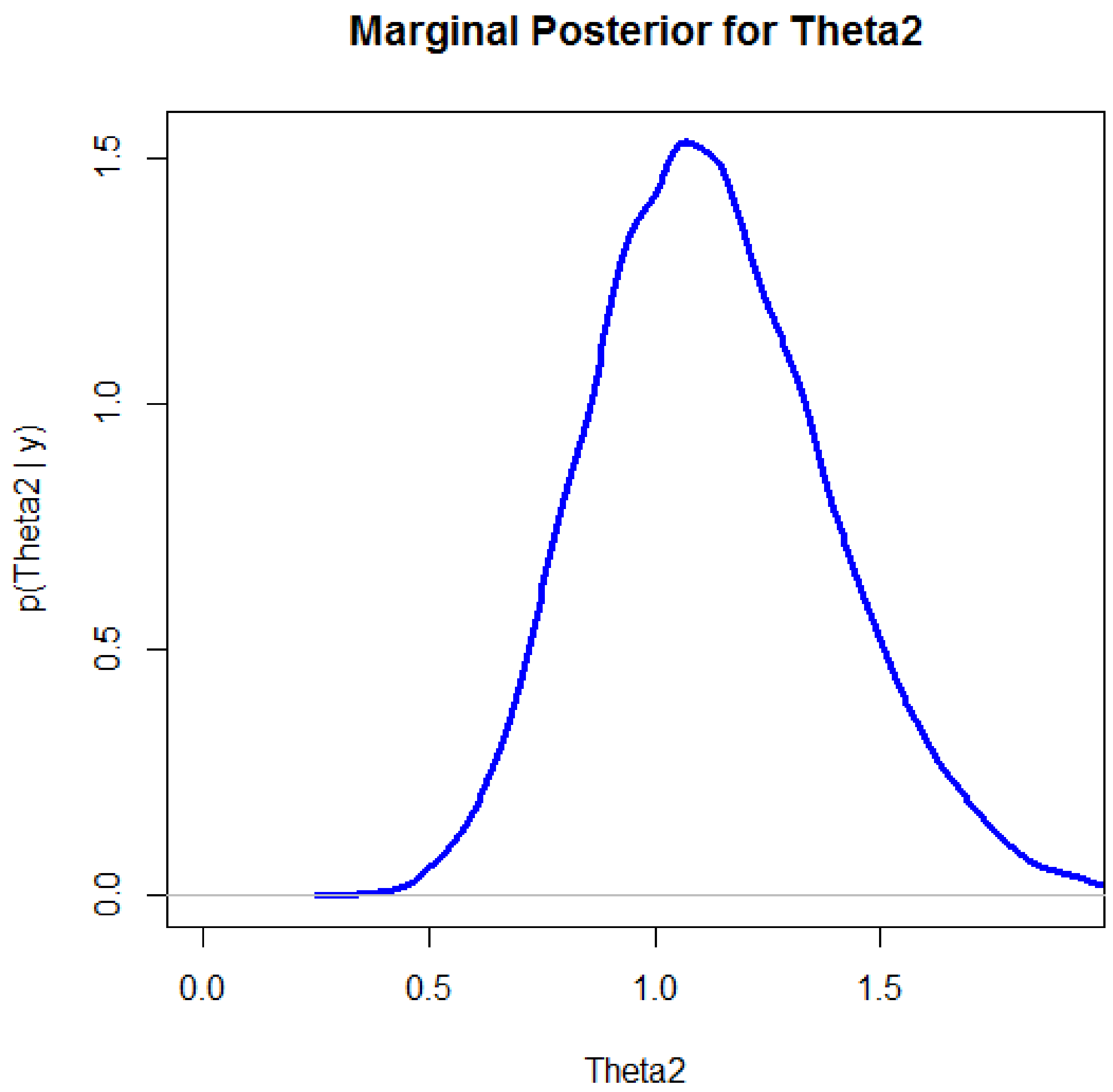
$a2 = 10$; $b2 = 0.1$ ⇒ $E(\theta_2) = 1\,; \; Var.(\theta_1) = 0.1$

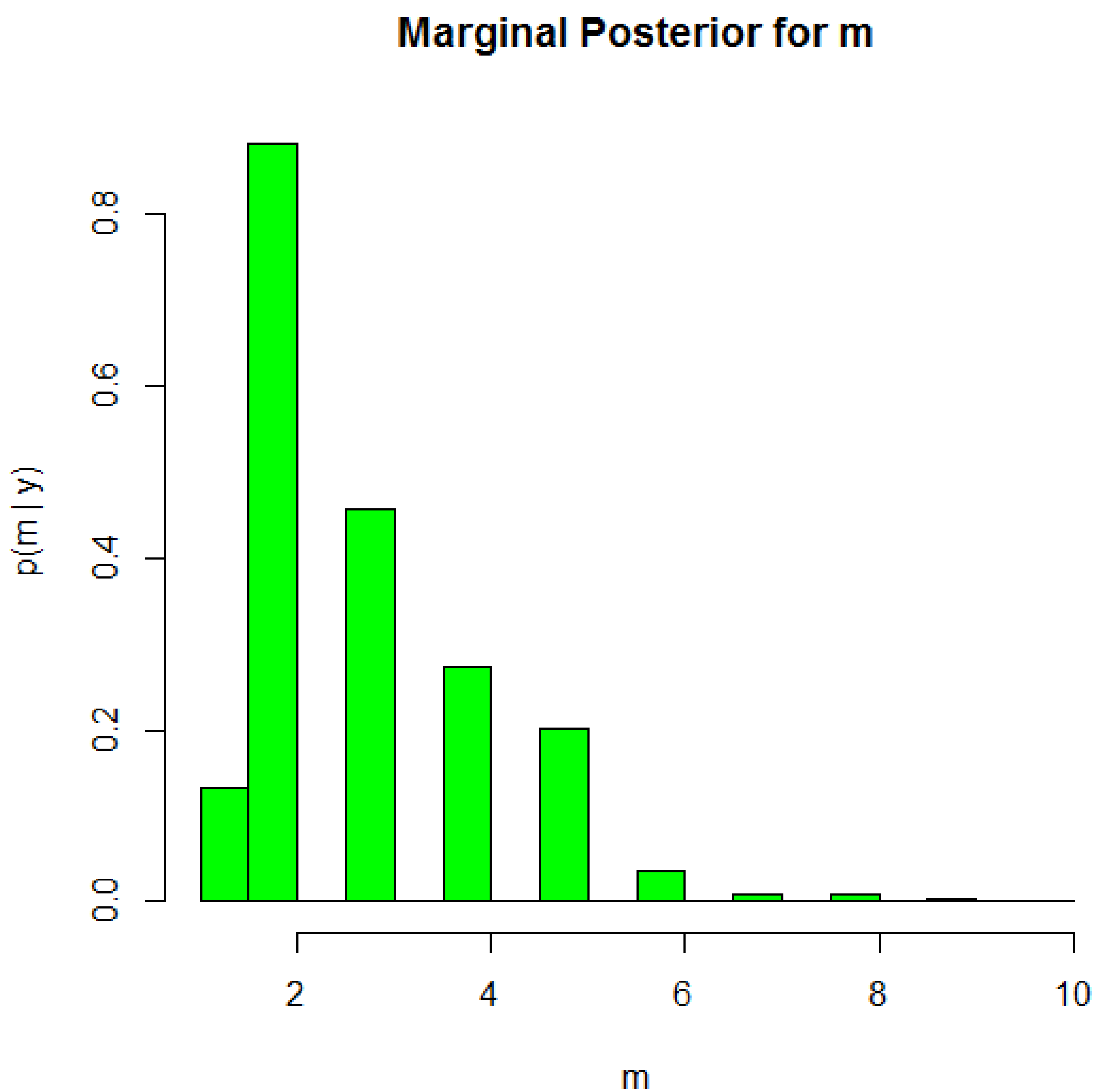
- Look at "Trace Plots" after the Burn-in, and the marginal posterior distributions:


Trace for Theta1
Theta1
8
7
6
5
4
3
0
10000
20000
30000
40000
Replications


Trace for Theta2
Theta2
Replications
2.5
2.0
1.5
1.0
0.5
0
10000
20000
30000
40000


Marginal Posterior for Theta1
p(Theta1 | y)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
3
4
5
6
7
8
Theta1


Marginal Posterior for Theta2
p(Theta2 | y)
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
Theta2



1

- Summary statistics for the marginal posterior distributions:

```
> summary(margm[(burnin+1):nrep])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   2.000   2.000   2.862   4.000  10.000 
> summary(margt1[(burnin+1):nrep])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  2.481   4.128   4.542   4.577   4.991   7.867 
> summary(margt2[(burnin+1):nrep])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.3343  0.9399  1.1100  1.1290  1.2990  2.4750 
```

m

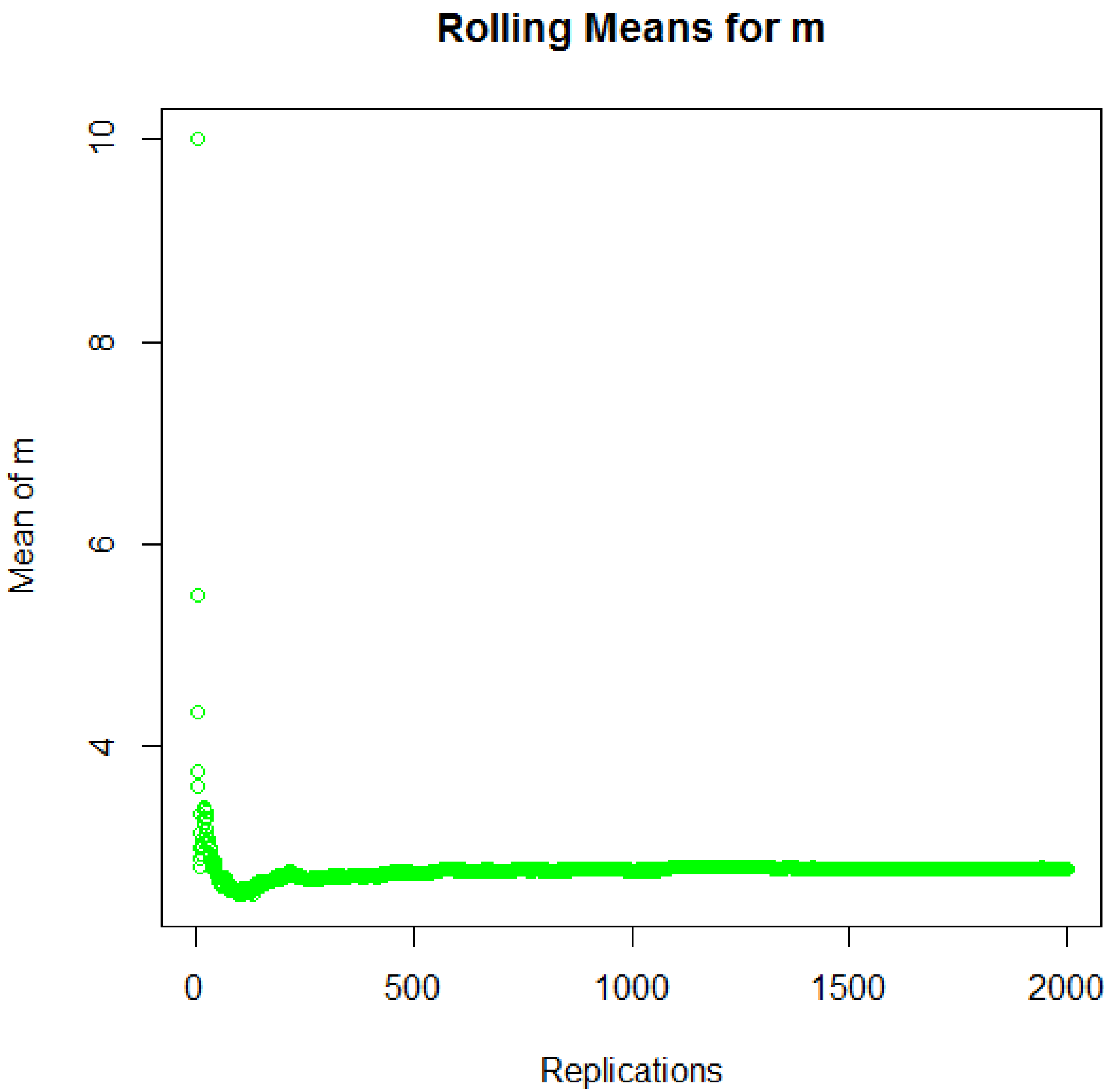
θ_1

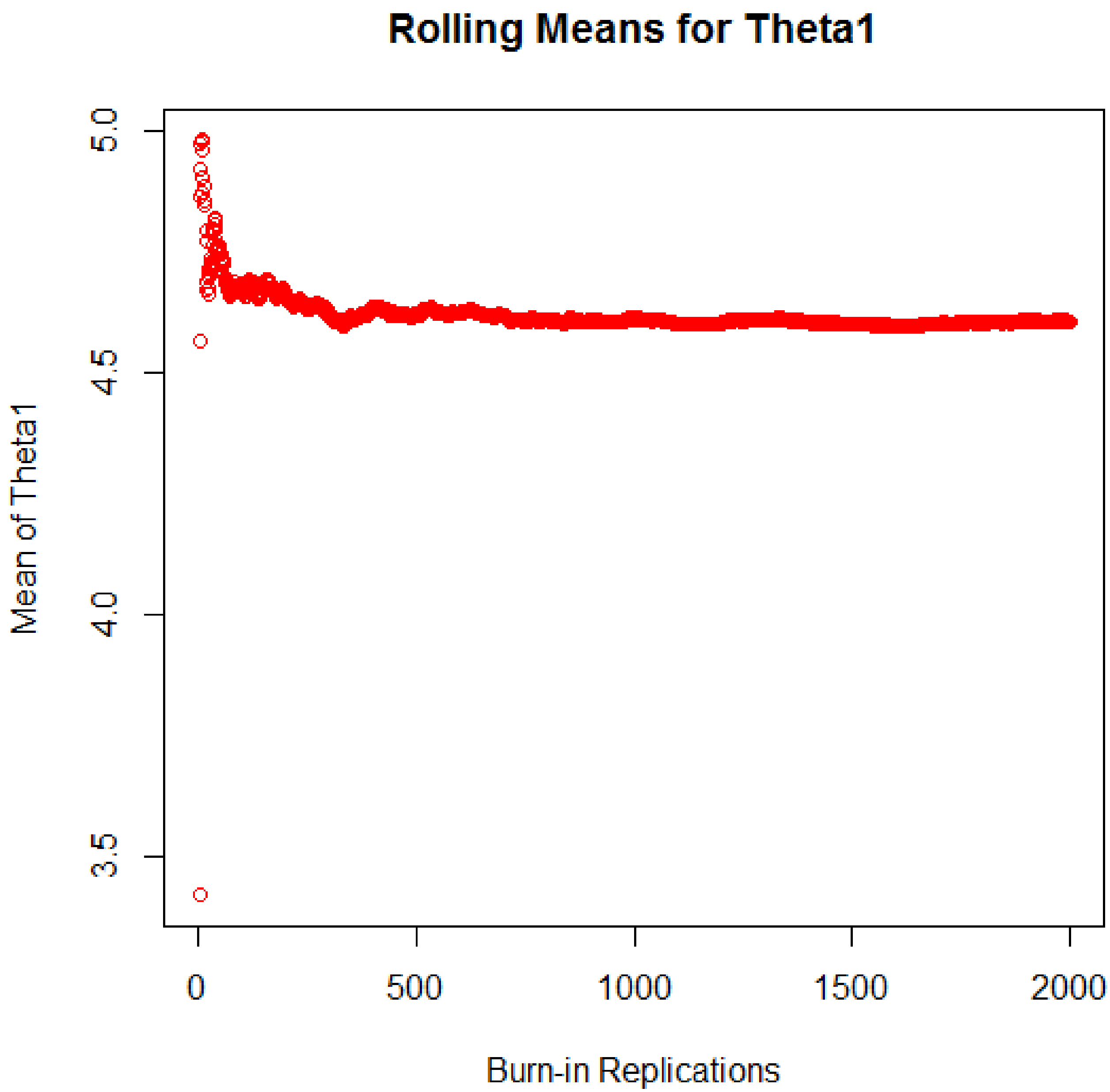
θ_2

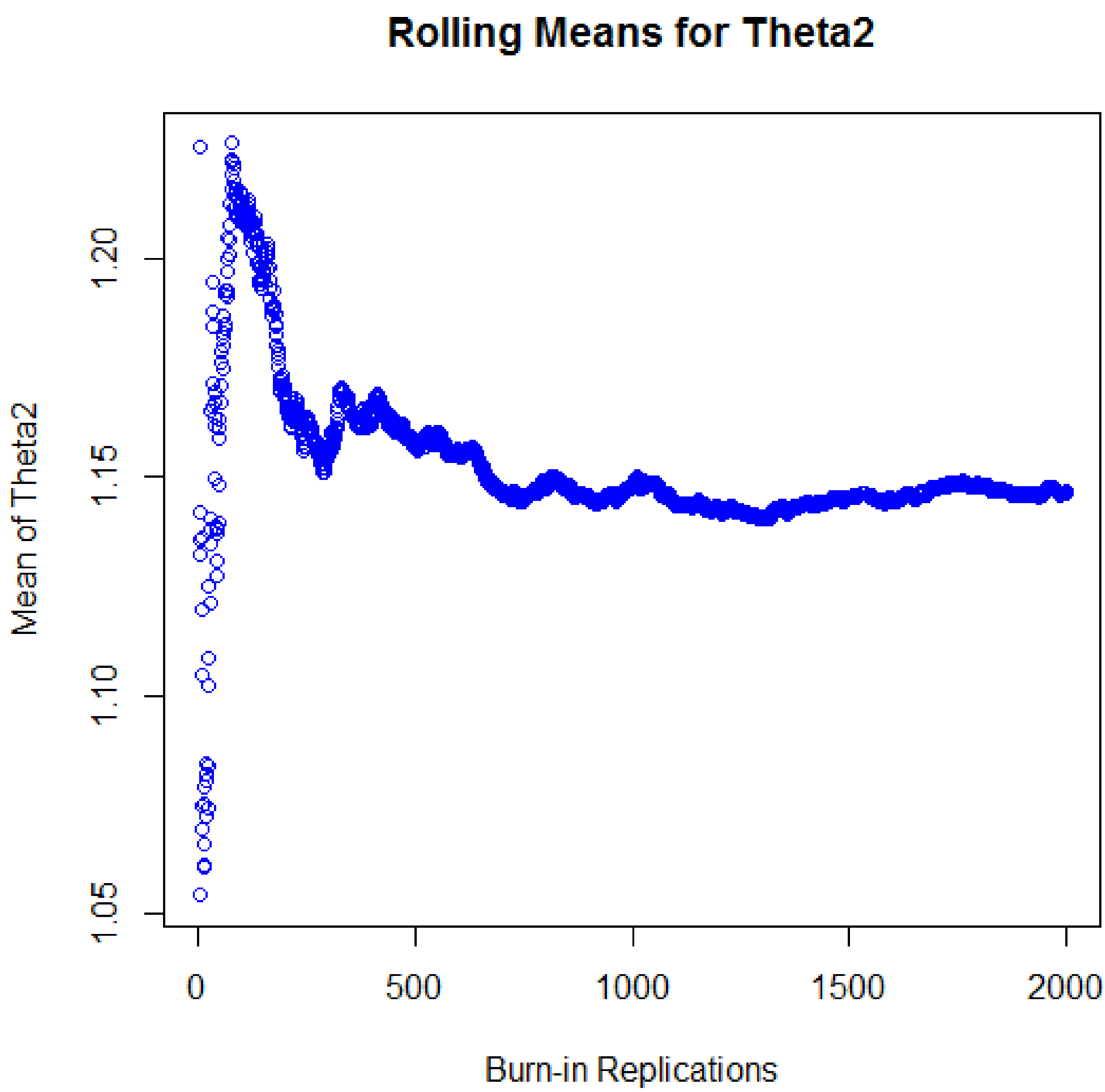
```
> modes<- c(mlv(margm[(burnin+1):nrep],method=mfv)[[1]] ,
+ mlv(margt1[(burnin+1):nrep], method="shorth")[[1]] ,
+ mlv(margt2[(burnin+1):nrep], method="shorth")[[1]])
> modes
[1] 2.000000 4.483419 1.072056
```

```
> quantile(margm[(burnin+1):nrep],  probs = c(1, 2.5, 5, 10, 90, 95, 97.5, 99)/100)
   1%  2.5%    5%   10%   90%   95% 97.5%   99% 
    1     1     1     2     5     5     6     6 
> quantile(margt1[(burnin+1):nrep],  probs = c(1, 2.5, 5, 10, 90, 95, 97.5, 99)/100)
      1%     2.5%       5%      10%      90%      95%    97.5%      99% 
3.221983 3.409586 3.581780 3.781613 5.415878 5.680817 5.921909 6.218699 
> quantile(margt2[(burnin+1):nrep],  probs = c(1, 2.5, 5, 10, 90, 95, 97.5, 99)/100)
       1%      2.5%        5%       10%       90%       95%     97.5%       99% 
0.5828191 0.6543828 0.7215368 0.7979071 1.4830775 1.5989234 1.7031821 1.8234375 
>
```

- Was the "Burn-in" period long enough?
- We used 2,000 drawings for the burn-in.
- Compute the "rolling means" of the draws over the burin-in period.
- That is, we're plotting what should start to become the means of the marginal posterior distributions if the burn-in is long enough.
- These should stabilize in value before the end of the burn-in period.
- Further diagnostics can be undertaken using the '**coda**' and '**mcmcplots**' packages in R.


Rolling Means for m
Mean of m
10
8
6
4
0
500
1000
1500
2000
Replications


Rolling Means for Theta1
Mean of Theta1
5.0
4.5
4.0
3.5
0
500
1000
1500
2000
Burn-in Replications


Rolling Means for Theta2
Mean of Theta2
1.05
1.10
1.15
1.20
0
500
1000
1500
2000
Burn-in Replications