Ma 322: Biostatistics
Homework Assignment 6

Prof. Wickerhauser

Read Chapter 11, "Algorithms for MCMC," pages 185–209 of our text.

Note: Although our text has no index or table of contents, it is easy to locate words in the electronic version using the Find function of your favorite PDF reader.

1. Suppose that K is a positive constant and $f(x) = Ke^{-x}$ is a pdf on $[1, \infty)$.

   (a) Find K. (Hint: Use Macsyma.)

   (b) Find the cumulative distribution function (cdf) of f.

   (c) Find the inverse cdf of f.

   (d) Use the formula in part c to simulate two samplings of 1000 values from the pdf f. Plot the resulting histograms over 30 equal-width bins. Use random number seeds 12345 and 6789, respectively, for the two samplings.

   Solution: Welcome back to calculus.

   (a) Since f is a pdf on $[1, \infty)$ it must satisfy

   $$1 = \int_1^\infty f(x)\,dx = K\left[-e^{-x}\right]|_1^\infty = Ke^{-1}.$$

   Conclude that $K = e$. Thus $f(x) = e \times e^{-x} = e^{1-x}$.

   Note: try the `maxima` command `integrate(exp(-x),x,1,inf);`.

   (b) The cumulative distribution function (cdf) of f is

   $$F(y) \stackrel{\text{def}}{=} \int_1^y f(x)\,dx = 1 - e^{1-y}, \qquad y \geq 1,$$

   with $F(y) = 0$ for $y < 1$.

   Note: the `maxima` commands `f:exp(1-x); integrate(f,x,1,y);` returns the formula for $y \geq 1$.

   (c) Set $z = F(y)$ and solve for y in terms of z by algebra:

   $$z = 1 - e^{1-y} \;\Rightarrow\; y = 1 - \log(1 - z).$$

   Note: the `maxima` command `solve(z= 1-exp(1-y), y);` returns three formulas, of which the unique real-valued formula should be used.

   (d) Implement the sampler with an argument allowing a choice of random number seed:

```
icdf<-function(n=1000, seed=NULL) { if( !is.null(seed) ) set.seed(seed);
z<-runif(n); y<-1-log(1-z); hist(y,30); }
```

The following R commands produce the required histograms in printable form:

```
pdf("hist6-1.pdf"); par(mfrow=c(1,2));
icdf(seed=12345); icdf(seed=6789); dev.off();
```

They are plotted at the end of this solution set. Note that most of the samples are close to 1 with a few taking large values. Still, this pdf is relatively"heavy-tailed" compared to the normal density. □

2. Consider the pdf $f(x) = K\sqrt{x}$ defined on the probability space $X = [0, 1]$.

   (a) Find the value of K.

   (b) Implement a rejection sampler for f and plot the histograms, in 20 bins, of two runs that each produce at least 1000 samples. Use random seeds 12345 and 6789, and count how many samples you actually get in each run.

   Solution: (a) Integrate f and the use the total area condition:

$$1 = \int_X f(x)\,dx = \int_0^1 K\sqrt{x}\,dx = \frac{2}{3}K \quad \Rightarrow K = \frac{3}{2}.$$

   (b) We expect about 1/3 of the samples to be rejected, so to get at least 1000 samples we should ask for more, say 2000 since computation is cheap.

```
rejsamp<-function(n=2000, seed=NULL) {
 if( !is.null(seed) ) set.seed(seed);  # use a seed if provided
 u<-runif(n); x<-runif(n); y<-x[(3/2)*sqrt(x)>(3/2)*u];  # rejection test
 hist(y,20, prob=TRUE); points(y,(3/2)*sqrt(y));  # plot histogram and pdf
 length(y);}  # publish how many samples were retained
```

   The last command prints the number of samples that survived the rejection step; it must be more than 1000. The following R commands produce the required histograms in printable form, and shows that more than 1000 samples were retained:

```
pdf("hist6-2.pdf"); par(mfrow=c(1,2));
rejsamp(seed=12345); rejsamp(seed=6789); dev.off();
```

   They are plotted at the end of this solution set. The actual density is plotted with points, and the `prob=TRUE` parameter in `hist()` scales the vertical axis to allow comparison. □

3. Suppose that X and Y are discrete random variables that have the following joint pdf:

   Joint pdf of X and Y

$f(X,Y)$	$Y=1$	$Y=2$	$Y=3$	$Y=4$
$X=1$	0.06	0.11	0.13	0.20
$X=2$	0.21	0.14	0.10	0.05

(a) Compute the marginal pdfs f_X and f_Y.

(b) Compute the complete conditional pdfs $f(X|Y)$ and $f(Y|X)$.

(c) Implement a Gibbs sampler for this joint pdf and simulate taking 1000 samples. Print the resulting matrix of counts as well as the normalized matrix that approximates the joint pdf. (Hint: see the example `R` code on p.193 of our text.)

(d) Compute the marginal pdfs for X and Y from the simulation in part c, giving both the raw counts and the normalized vectors that approximate f_X and f_Y.

Solution: Use the following `R` code:

```
f<-matrix(c(0.06,0.21,0.11,0.14,0.13,0.10,0.20,0.05),2,4);
fX<-rowSums(f); fY<-colSums(f);  # marginal pdfs
 fYX<-diag(1/fX)%*%f; fXY<-f%*%diag(1/fY); # conditional pdfs
 X<-1:2; Y<-1:4; x<-1; y<-1; sim<-matrix(0,2,4); # initialize
for(i in 1:1000) {
x<-sample(X,1,prob=fXY[,y]);
y<-sample(Y,1,prob=fYX[x,]);
sim[x,y]<-sim[x,y]+1;
}
sim; sim/sum(sim);  # approximate joint pdf
rowSums(sim); rowSums(sim/sum(sim));  # X marginal pdf
colSums(sim); colSums(sim/sum(sim));  # Y marginal pdf
```

This produces:

(a) $f_X = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$, $f_Y = (\,0.27 \quad 0.25 \quad 0.23 \quad 0.25\,)$.

(b)

$$f(X|Y) = \begin{pmatrix} 0.2222 & 0.44 & 0.5652 & 0.80 \\ 0.7778 & 0.56 & 0.4348 & 0.20 \end{pmatrix}; \qquad f(Y|X) = \begin{pmatrix} 0.12 & 0.22 & 0.26 & 0.40 \\ 0.42 & 0.28 & 0.20 & 0.10 \end{pmatrix}.$$

(c) Counts, estimated joint pdf:

$$\texttt{sim} = \begin{pmatrix} 64 & 112 & 117 & 209 \\ 185 & 177 & 78 & 58 \end{pmatrix}; \qquad \texttt{sim/sum(sim)} = \begin{pmatrix} 0.064 & 0.112 & 0.117 & 0.209 \\ 0.185 & 0.177 & 0.078 & 0.058 \end{pmatrix}.$$

(d) Row counts, estimated X marginal pdf:

$$\texttt{rowSums(sim)} = \begin{pmatrix} 502 \\ 498 \end{pmatrix}; \qquad \texttt{rowSums(sim/sum(sim))} = \begin{pmatrix} 0.502 \\ 0.498 \end{pmatrix}.$$

Column counts, estimated Y marginal pdf:

$$\begin{aligned} \texttt{colSums(sim)} &= (\,249 \quad 289 \quad 195 \quad 267\,); \\ \texttt{colSums(sim/sum(sim))} &= (\,0.249 \quad 0.289 \quad 0.195 \quad 0.267\,). \end{aligned}$$

□

4. Suppose that X and Y are continous random variables on $[0,1] \times [0,1]$ with the joint pdf

$$f(x,y) = c(1-x)(1-xy)(1-y),$$

where c is a positive constant.

(a) Compute c.

(b) Compute the marginal densities $f_X(x)$ and $f_Y(y)$.

(c) Are X and Y independent?

(d) Use the formulas from part b to compute the conditional pdfs $f_{X|Y}(x,y)$ and $f_{Y|X}(x,y)$.

(Hint: use Macsyma or similar software.)

Solution:

(a) Use Macsyma:

```
f: (1-x)*(1-x*y)*(1-y); integrate(integrate(f,x,0,1),y,0,1);
```

This gives $2/9$, so to have $\iint f(x,y)\,dxdy = 1$ requires $c = 9/2$.

(b) Use Macsyma:

```
c:9/2; fx:integrate(c*f,y,0,1); fy:integrate(c*f,x,0,1);
```

This gives

$$f_X(x) = \int_0^1 f(x,y)\,dy = \frac{3}{4}(1-x)(3-x) \qquad f_Y(y) = \int_0^1 f(x,y)\,dx = \frac{3}{4}(1-y)(3-y).$$

(c) X and Y are not independent, since their joint pdf is not the product of the marginals. To prove that, since all three are continuous functions, it is enough to check that they disagree at one point. Zero is the obvious choice:

$$f_X(0) = 9/4; \quad f_Y(0) = 9/4; \qquad f(0,0) = 9/2 \neq 81/16 = (9/4)^2.$$

(d) Use Macsyma with the formulas from part b to compute the conditional probabilities $f(X = x|Y = y) = f(x,y)/f_Y(y)$ and $f(Y = y|X = x) = f(x,y)/f_X(x)$:

```
fxy:c*f/fy; fyx:c*f/fx;
```

This gives

$$f(x|y) = \frac{6(1-x)(1-xy)}{(3-y)}; \qquad f(y|x) = \frac{6(1-y)(1-xy)}{(3-x)}.$$

□

5. (a,b) Implement the Gibbs sampler `gibbsBVN()` on p.195 of our text and use it to generate 1000 mean-$(0,0)$ samples with (a) $\rho = 0.33$ and (b) $\rho = -0.88$. Show your results in scatterplots.

(c,d) Use `mvrnorm()` to generate 1000 mean-$(0,0)$ samples from bivariate normal densities with covariances corresponding to $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ with (c) $\rho = 0.33$ and (d) $\rho = -0.88$. Show your results in scatterplots.

Solution: Use the following `R` code:

```
gibbsBVN <- function(x,y, n=1000, rho) {
m<-matrix(ncol=2,nrow=n);# create a matrix to store values
m[1,]<-c(x,y);           # initialize the first row
for (i in 2:n) {        # sampling iteration loop
 x<-rnorm(1,rho*y,sqrt(1-rho**2));#update x conditional on y
 y<-rnorm(1,rho*x,sqrt(1-rho**2));#update y conditional on new x
 m[i,]<-c(x,y); }       #store values in the matrix
m }  # return the result
pdf("gibbsbvn.pdf");  par(mfrow=c(2,2));
plot(gibbsBVN(0,0, rho=0.33)); plot(gibbsBVN(0,0,rho=-0.88));
require(MASS); m<-c(0,0); sr<-function(rho){matrix(c(1,rho,rho,1),2,2);}
plot(mvrnorm(1000,mu=c(0,0),Sigma=sr(0.33)));
plot(mvrnorm(1000,mu=c(0,0),Sigma=sr(-0.88)));
dev.off();
```

Note that the Gibbs sampler produces a mean=$(0,0)$ sampling in all cases, but the starting point (x,y) for its random walk must be specified. We choose that starting point to be the requested mean, $(0,0)$, in both cases. By contrast, the mean must be specified as the parameter `mu` given to `mvrnorm()`.

The results are printed in Figure HW6,Ex5. □

6. Implement the Metropolis algorithm as on p.203 of our text, only using an experimental outcome of 4 successes out of 9 trials as the likelihood, and a symmetric uninformative prior $f(\theta) \propto 1$, which is a beta density with $\alpha = 1, \beta = 1$ and thus a mean of $1/2$. Start with initial $\theta = 0.04$ as in the text's example, perform 1000 steps, and plot the histogram of the result after a 50-step burn-in against the known posterior beta-density having $\alpha = 5$ and $\beta = 6$.

   Solution: The `R` code on p.203 may be reused if the values of `y` and `n` are set to 4 and 9, respectively:

```
nchain<-1000; y<-4; n<-9; theta<-vector(length=nchain); theta[1]<-0.04;
for(i in 2:nchain){
thetastar<-runif(1); u<-runif(1);
r<-thetastar**y*(1-thetastar)**(n-y)/(theta[i-1]**y*(1-theta[i-1])**(n-y));
ifelse( u<r, theta[i]<-thetastar, theta[i]<-theta[i-1] );
}

pdf("mh56hist.pdf"); hist(theta[51:nchain],breaks=25,prob=T);
xx<-(1:100)/100; lines(xx,dbeta(xx,5,6)); dev.off();
```

   See the output in Figure HW6,Ex6. □

7. Implement the Metropolis-Hastings algorithm by modifying the code on p.203 of our text, again using an experimental outcome of 4 successes out of 9 trials as the likelihood, but with the nonsymmetric prior $f(\theta) \propto \theta^2$, which is a beta density with $\alpha = 3, \beta = 1$ and thus a mean of $3/4$. Start with initial $\theta = 0.04$ as in the text's example, perform 1000 steps, and plot the histogram of the result after a 50-step burn-in against the known posterior beta-density having $\alpha = 7$ and $\beta = 6$.

   Solution: The Metropolis-Hastings algorithm uses the prior density as the proposal sampler, then uses the likelihood ratio for rejection sampling. Hence we must modify the p.203 code by replacing `thetastar<-runif(1)` with `thetastar<-rbeta(1,alpha,beta)`, after assigning `alpha` and `beta` the respective values 3,1:

```
nchain<-1000; y<-4; n<-9; alpha<-3; beta<-1;
theta<-vector(length=nchain); theta[1]<-0.04;
for(i in 2:nchain){
thetastar<-rbeta(1,alpha,beta); u<-runif(1);
r<-thetastar**y*(1-thetastar)**(n-y)/(theta[i-1]**y*(1-theta[i-1])**(n-y));
ifelse( u<r, theta[i]<-thetastar, theta[i]<-theta[i-1] );
}
pdf("mh76hist.pdf"); hist(theta[51:nchain],breaks=25,prob=T);
xx<-(1:100)/100; lines(xx,dbeta(xx,7,6)); dev.off();
```

See the output in Figure HW6,Ex7 below. □

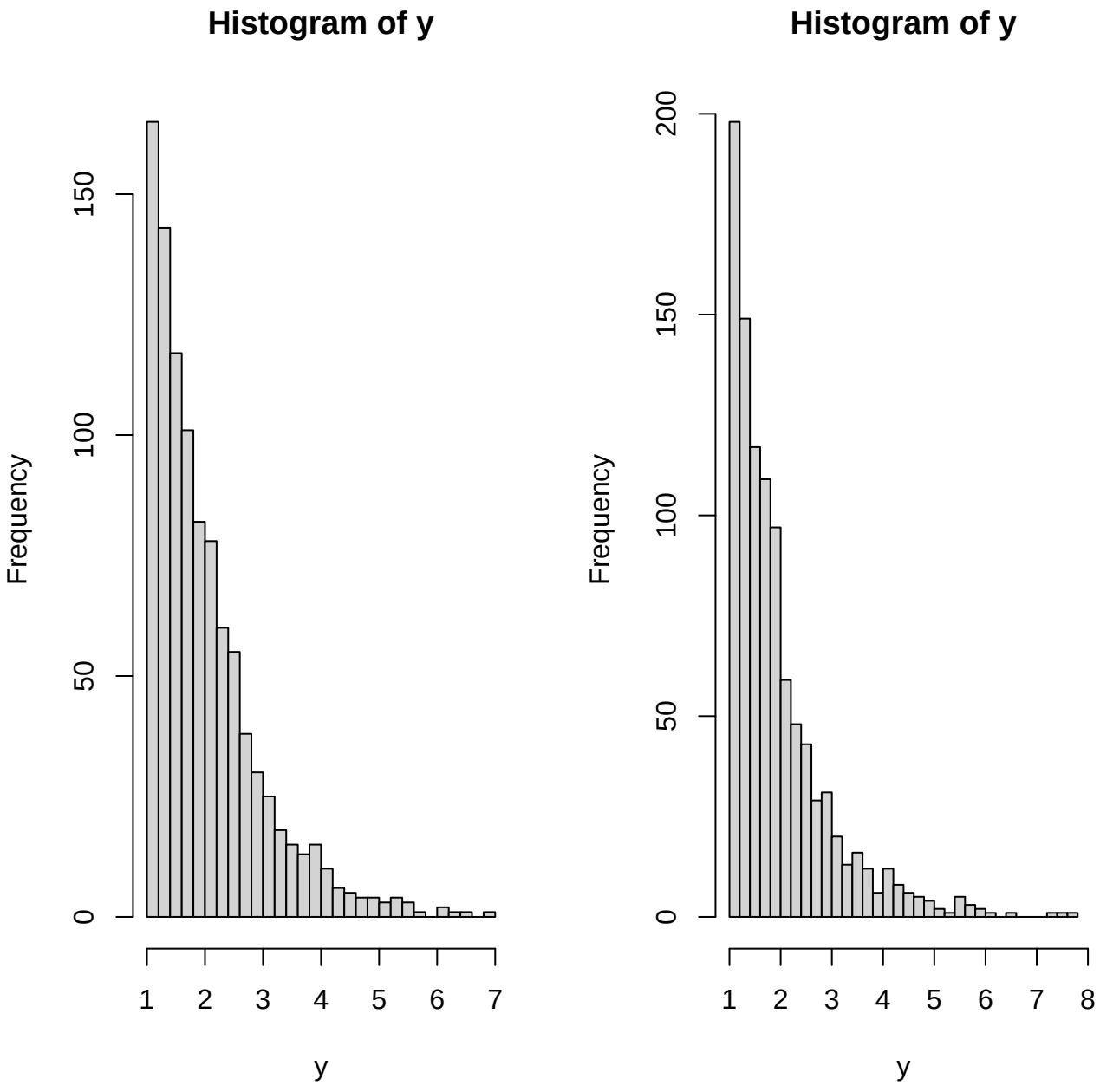

Figure 1: HW 6, Ex.1: Two histograms of the sampled pdf.

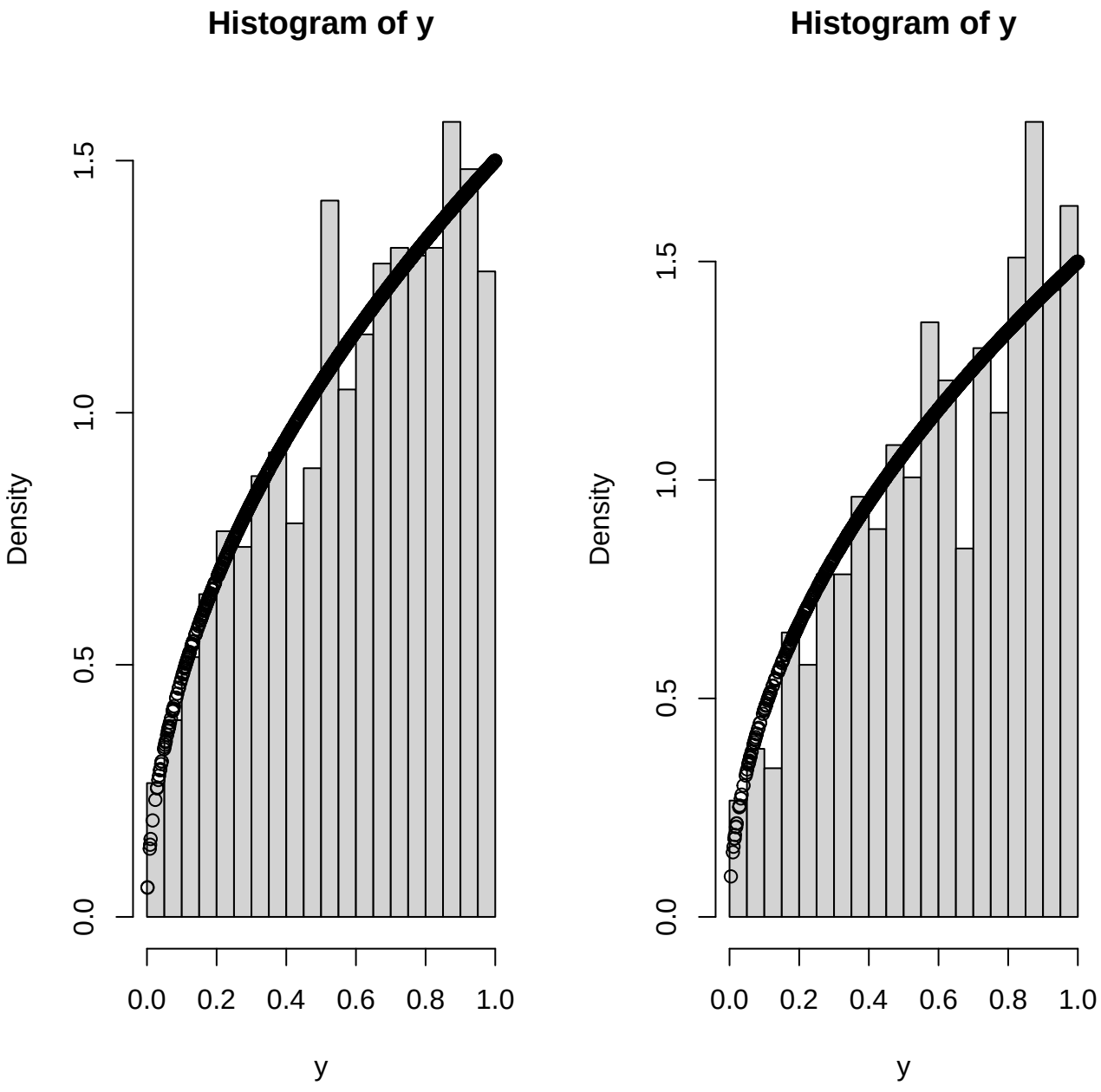

Figure 2: HW 6, Ex.2: Histograms of the rejection-sampled pdf.

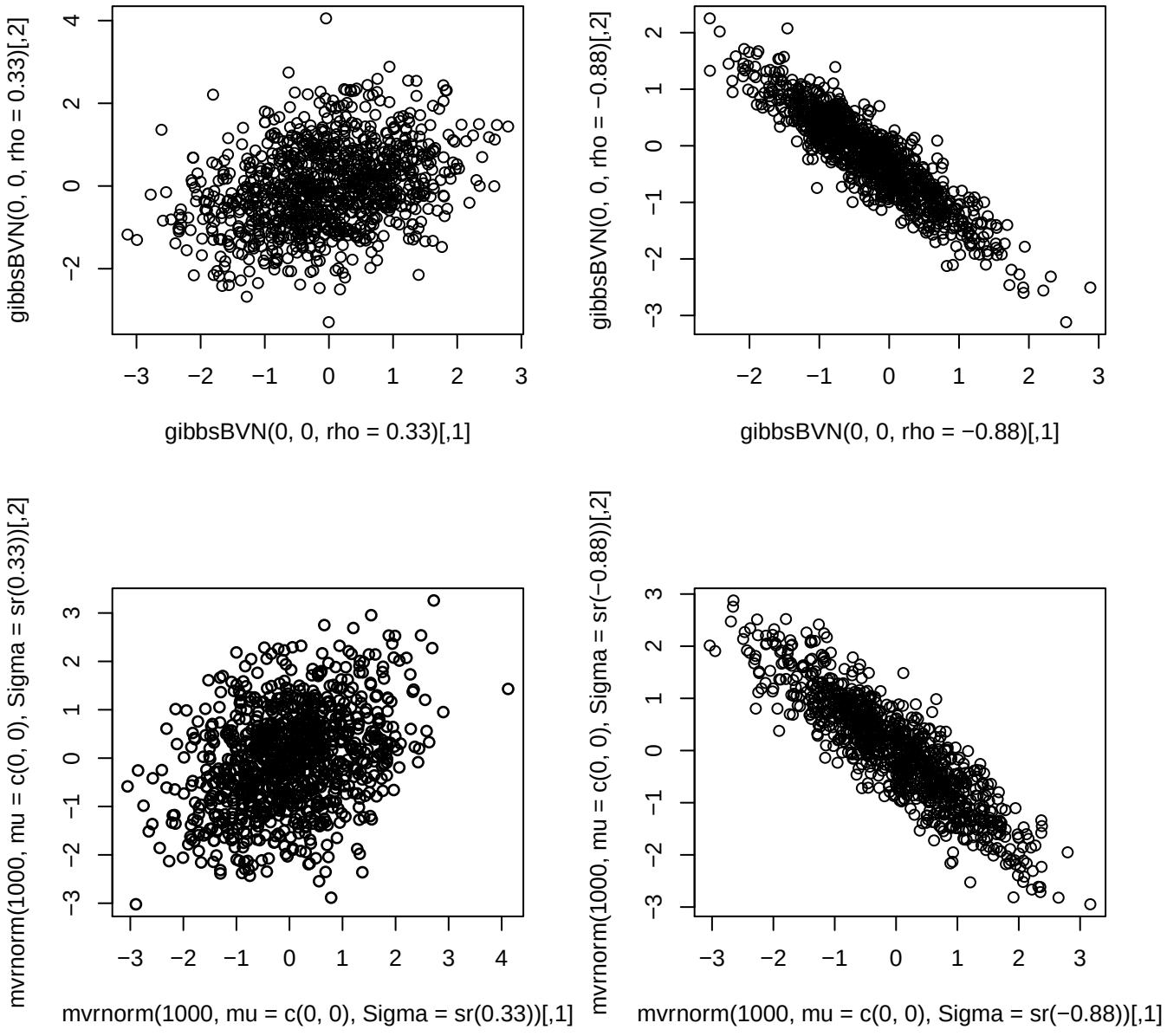

Figure 3: HW 6, Ex.5: Gibbs sampler for two bivariate normals, versus `mvrnorm()`.

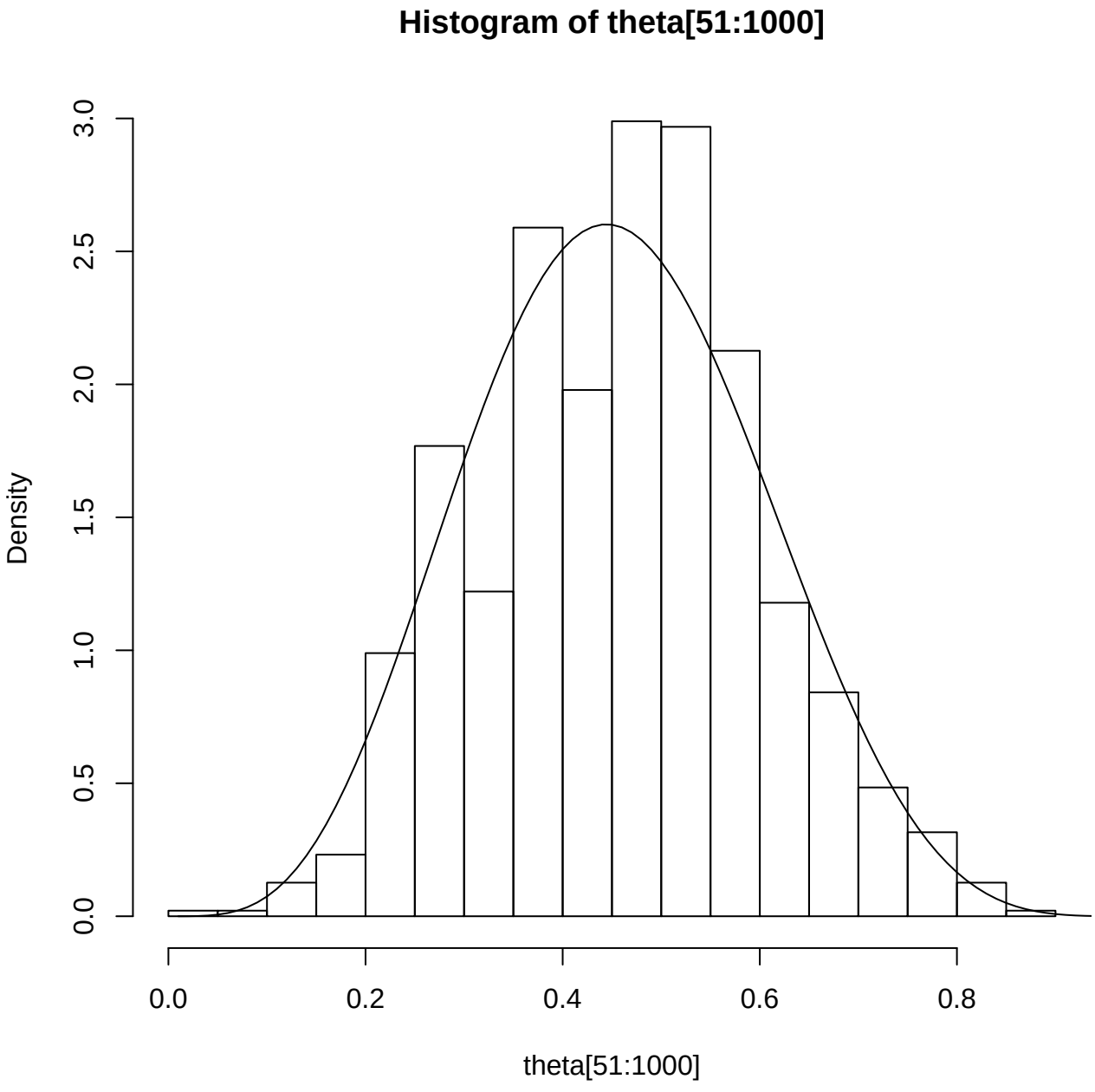

Figure 4: HW 6, Ex.6: Metropolis sampler for uniform prior, beta(5,6) posterior.

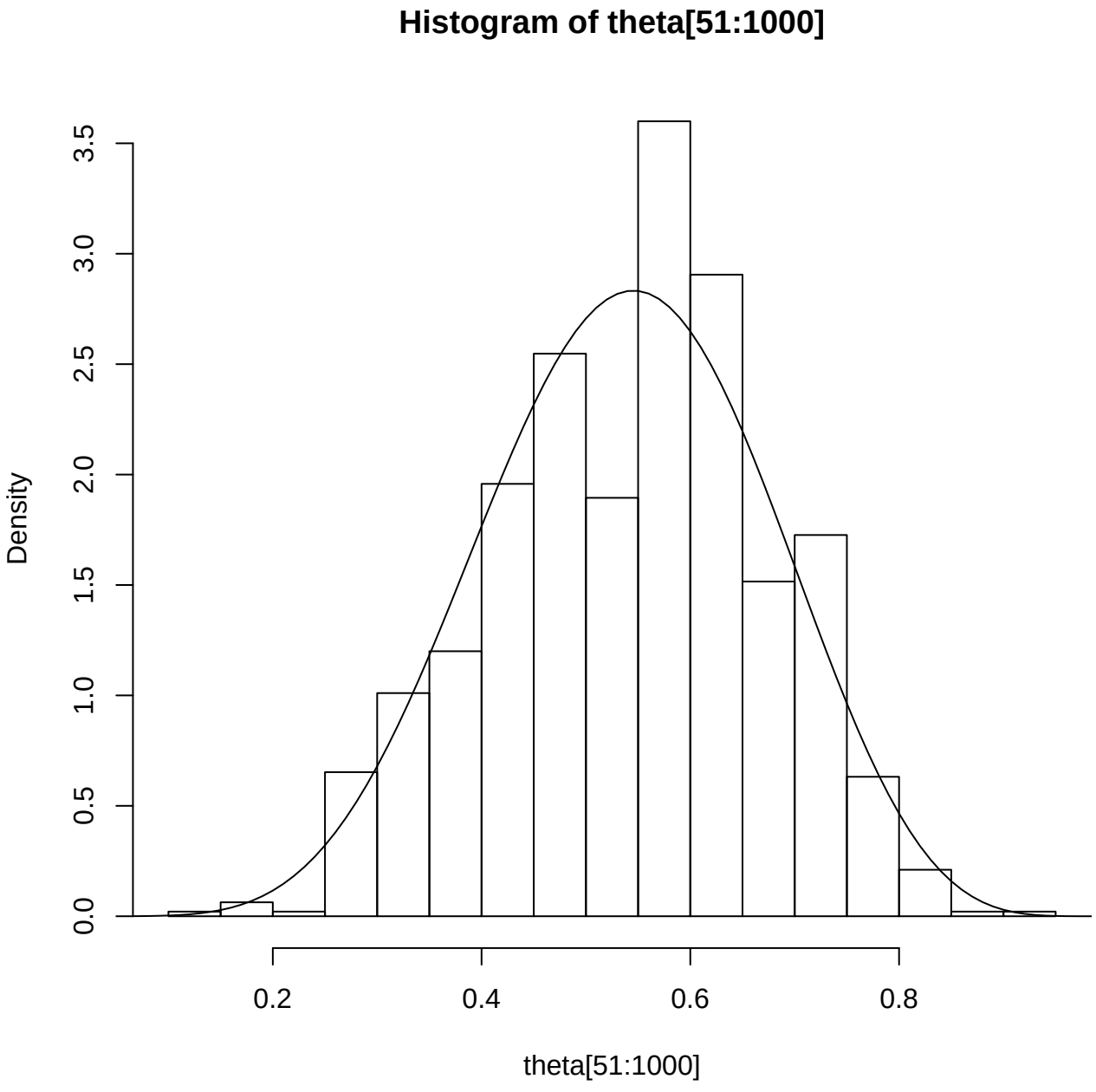

Figure 5: HW 6, Ex.7: Metropolis-Hastings sampler for beta (3,1) prior, beta(7,6) posterior.