## Problem 2

### (a)

> #a

> interexp<-read.table("interexp.dat",header=T)

> apply(interexp,2,mean,na.rm=T)

yA yB

24.20049 24.80535

> apply(interexp,2,var,na.rm=T)

yA yB

4.092800 4.691578

> cor(interexp[!(is.na(interexp$yA)|is.na(interexp$yB)),])

yA yB

yA 1.0000000 0.6164509

yB 0.6164509 1.0000000

The above R output suggests that

, and

### (b)

> mean.a.hat<-mean(interexp$yA,na.rm = T)

> mean.b.hat<-mean(interexp$yB,na.rm = T)

> var.a.hat<-var(interexp$yA,na.rm = T)

> var.b.hat<-var(interexp$yB,na.rm = T)

> cor.hat<-cor(interexp[!(is.na(interexp$yA)|is.na(interexp$yB)),])[1,2]

> interexp.b<-interexp

> interexp.b[,1]<-ifelse(is.na(interexp[,1]),mean.a.hat+(interexp[,2]-mean.b.hat)\*cor.hat\*sqrt(var.a.hat/var.b.hat),interexp[,1])

> interexp.b[,2]<-ifelse(is.na(interexp[,2]),mean.b.hat+(interexp[,1]-mean.a.hat)\*cor.hat\*sqrt(var.b.hat/var.a.hat),interexp[,2])

> t.test(interexp.b[,1],interexp.b[,2],paired = T)$conf.int

[1] -0.9850730 -0.2383347

attr(,"conf.level")

[1] 0.95

We see that the 95% confidence interval for from the t-test is (-0.9851, -0.2383).

### (c)

> mean(THETA[,1]-THETA[,2])

[1] -0.6042315

> quantile(THETA[,1]-THETA[,2],prob=c(0.025,0.975))

2.5% 97.5%

-0.6098562 -0.5977206

The posterior mean for is -0.6042, the posterior confidence interval is (-0.6099, -0.5977). The confidence interval is much narrower than that in part (b), and the mean difference estimated in (b) is not in the 95% confidence interval here. When estimating the mean, variance and covariance, we only use partial data in (b). The imputation in (c) uses more data we have. Hence the result in (c) is more reliable.

### (d)

The imputed values assumes that for each , and are independent. The assumption is inappropriate as each and are responses from the same experimental subject. Therefore, the imputation used in (b) would be closer to the real data.

## Problem 3

### (a)

We can use the MCMC quantiles to approximate the confidence interval of the marginal posterior distribution of .

> c.interval<-list()

> for(i in 1:7)

+ {

+ c.interval[[i]]<-rbind(quantile(THETA.n[,i],prob=c(.025,.5,.975)),quantile(THETA.d[,i],prob=c(.025,.5,.975)))

+ }

> c.interval

[[1]]

2.5% 50% 97.5%

[1,] 2.636795 2.923371 3.216014

[2,] 4.124898 4.700312 5.297282

[[2]]

2.5% 50% 97.5%

[1,] 107.5743 110.0230 112.5496

[2,] 138.5332 143.1252 147.7116

[[3]]

2.5% 50% 97.5%

[1,] 68.65971 69.90724 71.15769

[2,] 72.84043 74.72377 76.57940

[[4]]

2.5% 50% 97.5%

[1,] 26.21938 27.28752 28.34311

[2,] 31.44072 32.97868 34.49150

[[5]]

2.5% 50% 97.5%

[1,] 30.73134 31.42488 32.11092

[2,] 34.86775 35.81558 36.77776

[[6]]

2.5% 50% 97.5%

[1,] 0.4154752 0.4463986 0.4781099

[2,] 0.5585716 0.6165581 0.6745132

[[7]]

2.5% 50% 97.5%

[1,] 28.18203 29.20616 30.25589

[2,] 34.84185 36.42595 38.07126

For each, the 95% confidence interval of and does not overlap.

We resample from the posterior samples to approximate .

> #approximation

> theta.greater<-function(i){mean(sample(THETA.d[,i],10000,replace = T)>sample(THETA.n[,i],10000,replace = T))}

> p.greater<-NULL

> for(i in 1:7)

+ {

+ p.greater<-c(p.greater,theta.greater(i))

+ }

> p.greater

[1] 1 1 1 1 1 1 1

We find that for all .

### (b)

Below are the posterior means of and respectively.

> round(matrix(apply(SIGMA.d,2,mean),ncol=7,byrow=T),4)

[,1] [,2] [,3] [,4] [,5] [,6] [,7]

[1,] 15.3502 -9.9579 6.1345 -4.1370 -4.6443 -0.0943 23.4833

[2,] -9.9579 979.5164 33.3620 30.8732 10.2496 0.2410 34.5212

[3,] 6.1345 33.3620 157.0444 12.3488 18.0442 -0.1788 36.2490

[4,] -4.1370 30.8732 12.3488 107.9704 35.6182 0.5321 -7.5545

[5,] -4.6443 10.2496 18.0442 35.6182 43.7070 0.3861 -13.7376

[6,] -0.0943 0.2410 -0.1788 0.5321 0.3861 0.1588 -0.1474

[7,] 23.4833 34.5212 36.2490 -7.5545 -13.7376 -0.1474 117.3389

> round(matrix(apply(SIGMA.n,2,mean),ncol=7,byrow=T),4)

[,1] [,2] [,3] [,4] [,5] [,6] [,7]

[1,] 7.7632 4.5990 6.6520 3.6783 0.0147 -0.0414 18.2956

[2,] 4.5990 590.4298 56.2502 32.4324 25.4668 0.6602 42.9217

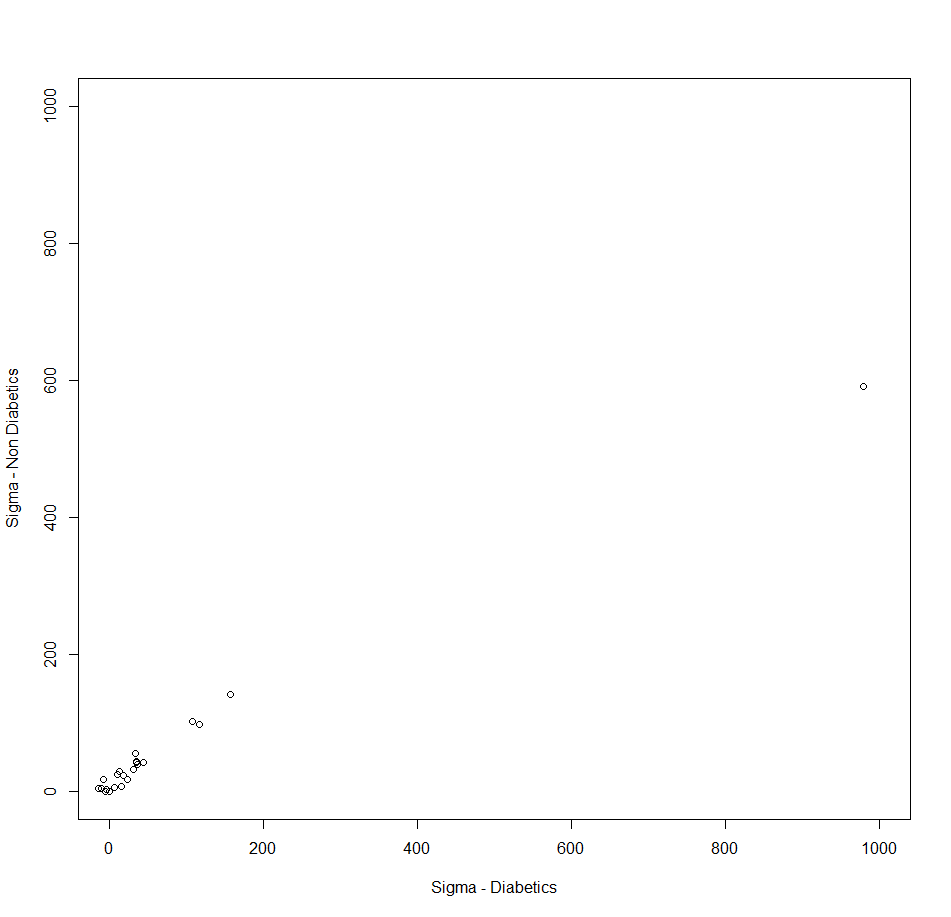
[3,] 6.6520 56.2502 141.7871 28.7730 23.1466 -0.1311 39.6099

[4,] 3.6783 32.4324 28.7730 101.6345 44.3158 0.0537 17.5128

[5,] 0.0147 25.4668 23.1466 44.3158 42.8710 0.0938 4.4759

[6,] -0.0414 0.6602 -0.1311 0.0537 0.0938 0.0892 0.0642

[7,] 18.2956 42.9217 39.6099 17.5128 4.4759 0.0642 98.1311

Here is a plot of posterior means of Sigma for both groups.

The value on x-axis and y-axis stands for the means of each element in posterior samples of and respectively.

We notice that the variance of each variable is smaller in the non-diabetics group, while the covariance between most variables in the non-diabetics group are larger than that in the diabetics group.

## Problem 4

### (a)

Referring to Hoff Section 9.2, we recognize it to be proportional to a MVN density

### (b)

As implies and implies

We know that such that and such that.

Let

We know

Which is a density constrained to the interval.

We know that, then

where

### (c)

The following code implement the Gibbs sampler and shows the effective sample sizes of unknown parameters.

> divorce<-read.table("divorce.dat")

> x<-divorce[,1]

> y<-divorce[,2]

> n<-nrow(divorce)

> tauB2<-tauC2<-16

> #Initial values

> beta<-0

> c<-0

> z<-NULL

> for(i in 1:n){

+ mu<-beta\*x[i]

+ if(y[i]==1) {u<-runif(1,pnorm(c-mu),1)}else{u<-runif(1,0,pnorm(c-mu))}

+ z<-c(z,mu+qnorm(u))

+ }

> C<-Z<-BETA<-NULL

> #

> beta.var<-1/(sum(x^2)+1/tauB2)

> for(i in 1:10000){

+ #update beta

+ beta<-rnorm(1,sum(divorce[,1]\*z)\*beta.var,sqrt(beta.var))

+ #update c

+ z.a<-max(z[y==0])

+ z.b<-min(z[y==1])

+ u<-runif(1,pnorm(z.a/sqrt(tauC2)),pnorm(z.b/sqrt(tauC2)))

+ c<-sqrt(tauC2)\*qnorm(u)

+ #update z

+ z<-NULL

+ for(i in 1:n){

+ mu<-beta\*x[i]

+ if(y[i]==1) {u<-runif(1,pnorm(c-mu),1)}else{u<-runif(1,0,pnorm(c-mu))}

+ z<-c(z,mu+qnorm(u))

+ }

+ #save result

+ BETA<-c(BETA,beta)

+ C<-c(C,c)

+ Z<-rbind(Z,z)

+ }

> library(coda)

> effectiveSize(BETA)

var1

726.6

> effectiveSize(C)

var1

406.301

> effectiveSize(Z)

var1 var2 var3 var4 var5

2302.2260 5801.3143 5566.8111 2945.2419 6187.2975

var6 var7 var8 var9 var10

9357.3111 1572.4932 940.8099 6327.4259 2258.4582

var11 var12 var13 var14 var15

7051.5427 1978.8407 1306.4323 1688.1580 1555.6341

var16 var17 var18 var19 var20

8399.8496 1646.0886 1951.1297 6399.3181 5249.7727

var21 var22 var23 var24 var25

1910.1720 5122.3126 1485.4331 813.5722 3529.1381

The effective sample sizes for β and c are 726.6 and 406.301 respectively.

The effective sample sizes for z1,…, z25 are on the output above, where the value under “vari” is the effective sample size for zi for i = 1,…, 25.

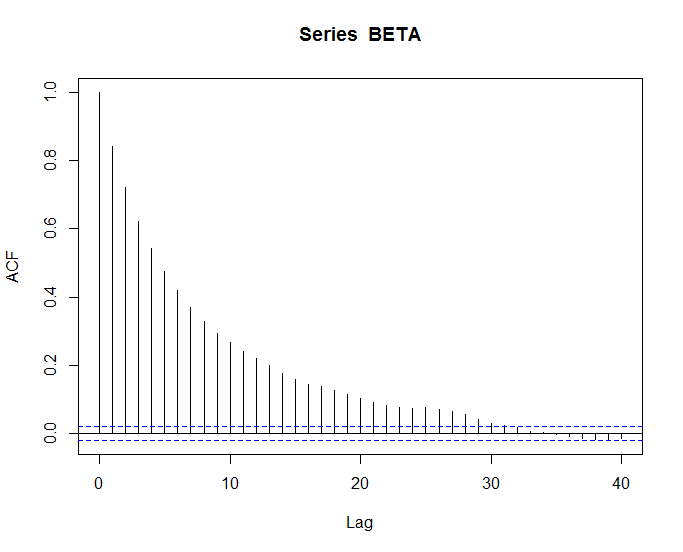
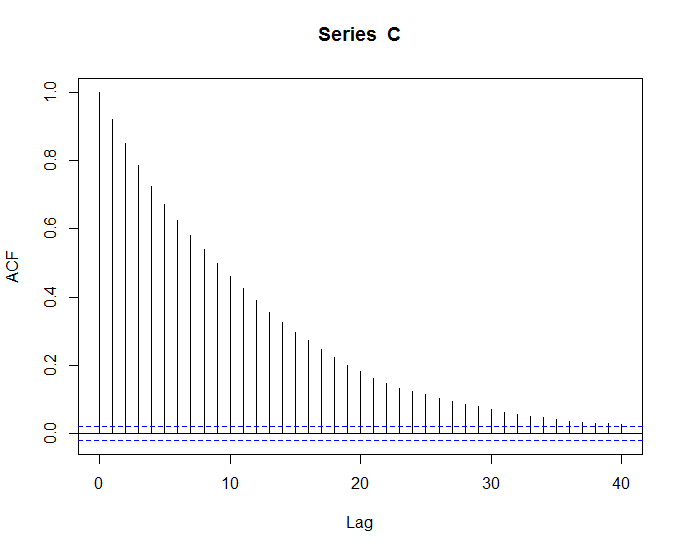
Below are the autocorrelation for the parameters:

Figure 1 acf(C)

Figure 2 acf(BETA)

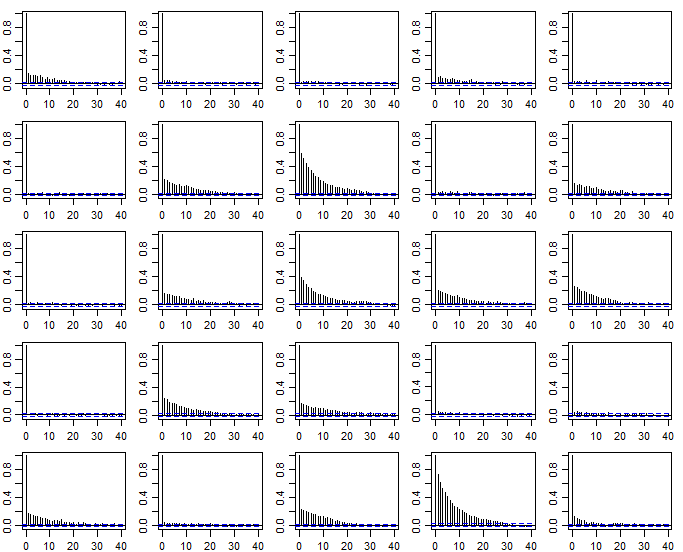


Figure 3 Autocorrelation for Z

In Figure 3, the graph on the ith row, jth column shows the autocorrelation for Z5(j-1)+i.

The mixing of the Markov chain is quite slow as we have relatively high autocorrelation and low effective sample sizes for β, c, and some Zis. The Gibbs sampler may need more iterations to have a better approximation.

### (d)

From the following R output, we can see that the 95% posterior confidence interval for β is (0.1071, 0.6686). And.

> quantile(BETA,probs = c(0.025,0.975))#95% confidence interval for beta

2.5% 97.5%

0.0921617 0.6283478

> length((BETA>0)==TRUE)/10000

[1] 1