รายการอ้างอิง

[1] Quinn, Kevin M.. 2004. Bayesian Factor Analysis for Mixed Ordinal and Continuous Response. Political Analysis 12(4) : 338-353.

[2] Cowles, M. K.. 1996. Accelerating Monte Carlo Markov Chain Convergence for Cumulative Link Generalized Linear Models. Statistics and Computing 6 : 101-111.

[3] Lovasz and Vempala. 2006. Fast Algorithm for Log-concave Functions: Sampling, Rounding, Integration and Optimization. Proceedings of the 47th IEEE Symposium on Foundations of Computer Science (FOCS'06) : 57-68.

[4] Chen M.-H and Schmeiser B.W.. 1993. Performance of the Gibbs, Hit-and-Run, and Metropolis Samplers. Journal of Computational and Graphical Statistics 12, 3 : 251-272.

[5] Geman S. and Geman D.. 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6 : 721-741.

[6] Smith, R. L. 1984. Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. Operations Research 6 : 1296-1308.

[7] Bélisle, C.J.P., Romeijn, H. E., and Smith, R. L.. 1993. Hit-and-Run algorithm for generating multivariate distributions. Mathematics of Operations Research 18 : 255-266.

[8] Romeijn H.E. and Smith R.L. 1994. Simulated annealing for constrained global optimization. Journal of Global Optimization 5 : 101-126.

[9] Flegal J.M., Haran M. and Jones G.L. 2008. Markov Chain Monte Carlo: Can We Trust the Third Significant Figure?. Statistical Science 23, 2 : 250-260.

ภาคผนวก

ภาคผนวก ก

โปรแกรมสำหรับงานวิจัย

โปรแกรม R สำหรับดำเนินงานตามขั้นตอนในการวิจัย

```
seednum<-101;
signvec<-c(-1,0,0,0,0);
roundnum<-500000;
numbatch<-20;
recname<-pedata[,1];
varname<-names(peframe);

#ดำเนินการด้วยการสุ่มตัวอย่างแบบกิบส์
set.seed(seednum);
Gibb2time<-system.time(Gibb2<-fmcmcG(peframe,signvec,roundnum,100));
outGibb2<-batchMeans(Gibb2$Record,numbatch);
Gibb2mean<-meanplot(Gibb2$Record,10);

#ดำเนินการด้วยการสุ่มตัวอย่างแบบฮิตแอนด์รัน
set.seed(seednum);
GHRtime<-system.time(GHR<-fmcmcGHR(peframe,signvec,roundnum,100));
outGHR<-batchMeans(GHR$Record,numbatch);
GHRmean<-meanplot(GHR$Record,10);
```

โปรแกรมสำหรับฟังก์ชัน fmcmcG

```
fmcmcG<-function(dframe,signvec,roundnum,batch){
levelvec<-sapply(sapply(dframe, levels),length);
varcode<-ifelse(levelvec==0,1,ifelse(levelvec==2,3,2));
numrec<-nrow(dframe);
numvar<-ncol(dframe);
recname<-row.names(dframe);
varname<-names(dframe);
burnin<-10000;

#Standardize continuous data
for(j in 1:numvar){
ifelse(varcode[j]==1,dframe[,j]<-(dframe[,j]-
        mean(dframe[,j]))/sd(dframe[,j]),dframe[,j]<-dframe[,j]);
}

#Prior Parameters
lparam<-matrix(0,numvar,2);
Lparam<-matrix(0.25,numvar,2);
aparam<-rep(0.001/2, length=numvar);
bparam<-rep(0.001/2, length=numvar);
t2param<-rep(0.01, length=numvar);

#Initialize gamma ordmax
gamma<-c();
for(j in 1:numvar){
if(varcode[j]!=2){
subgamma<-list(0);
}
else{
```

```
subgamma<-list(c(0,qnorm((seq(2:(levelvec[j]-1))/(2*levelvec[j])+0.5))));
}
gamma<-c(gamma,subgamma);
}

#Initialize xstar
xstar<-matrix(0,numrec,numvar);
for(j in 1:numvar){
if(varcode[j]==1) xstar[,j]<-dframe[,j];
if(varcode[j]==3) xstar[,j]<-as.numeric(dframe[,j])-1.5;
if(varcode[j]==2){
for(i in 1:numrec){
if(dframe[i,j]==1) xstar[i,j]<-0-0.5;
if((dframe[i,j]!=1)&(dframe[i,j]!=levelvec[j]))xstar[i,j]<-
        (0.5*(gamma[[j]][dframe[i,j]]+gamma[[j]][(as.numeric(dframe[i,j])-1)]));
if(dframe[i,j]==levelvec[j]) xstar[i,j]<-(gamma[[j]][(levelvec[j]-1)]+0.5);
}
}
}

#Initialize z
z<-rnorm(numrec);

#Initialize psi
psi<-rep(0,length=numvar);
psi<-ifelse(varcode==1,aparam*bparam,1);

#Initialize lambda constrain lambda2(courts) to be "-"
lambda<-matrix(0,numvar,2);
```

```
lambda[,1]<-ifelse(varcode==1,0,rnorm(numvar,lparam[,1],1 lambdatemp<-
        qnorm(runif(numvar),0,1);
lambda[,2]<-ifelse(signvec==-1,-
        abs(lambdatemp),ifelse(signvec==1,abs(lambdatemp),lambdatemp));

#Initialize output
lambda.col<-c();
gamma.col<-c();
## Get output column names
for(k in 1:ncol(lambda)){
lambda.col<-c(lambda.col,paste(varname,".l",k,sep=""));
}
for(j in 1:numvar){
if(levelvec[j]>2){
gamma.col<-c(gamma.col,paste(varname[j],".g",1:(levelvec[j]-1),sep=""));
}
}
outmat.col<-c(recname,lambda.col,gamma.col);
outmat<-
        matrix(0,nrow=roundnum,ncol=length(outmat.col),dimnames=list(c(1:roundnum),o
        utmat.col));

###Gibb Sampler
for(n in 1:(roundnum+burnin)){
outvec<-c();
gammavec<-c();
lambdavec<-c();

#Sample gamma
for(j in  1:numvar){
```

```
if(varcode[j]==2){
for(k in 2:(levelvec[j]-1)){
a<-max(max(xstar[,j][dframe[,j]==k]),gamma[[j]][k-1]);
if(k==(levelvec[j]-1)){
b<-min(xstar[,j][dframe[,j]==(k+1)])
}
else{
b<-min(min(xstar[,j][dframe[,j]==(k+1)]),gamma[[j]][k+1]);
}
gamma[[j]][k]<-runif(1,a,b);
}
gammavec<-c(gammavec,gamma[[j]]);
}#End if varcode[j]==2
}#End Sample Gamma[[j]]

#sample xstar for ordinal and binary variables
for(j in 1:numvar){
if(varcode[j]==2){
for(k in 1:levelvec[j]){
x<-xstar[dframe[,j]==k,j];
kk<-floor(runif(1,1,length(x)+1));
if(k==levelvec[j]){
rightnorm<-1;
}
else{
rightnorm<-pnorm(gamma[[j]][k],lambda[j,1]+lambda[j,2]*z[dframe[,j]==k][kk],1);
}
if(k==1){
leftnorm<-0;
}
```

```
else {
leftnorm<-pnorm(gamma[[j]][k-1],lambda[j,1]+lambda[j,2]*z[dframe[,j]==k][kk],1);
}
xstar[dframe[,j]==k,j][kk]<-qnorm(runif(1)*(rightnorm-
        leftnorm)+leftnorm,lambda[j,1]+lambda[j,2]*z[dframe[,j]==k][kk],1);
}
}
else{
if(varcode[j]==3){
for(i in 1:numrec){
A<-pnorm(0,lambda[j,1]+lambda[j,2]*z[i],1,lower.tail=TRUE);
if(dframe[i,j]==0){
xstar[i,j]<-qnorm(runif(1)*A,lambda[j,1]+lambda[j,2]*z[i],1);
}
else{
xstar[i,j]<-qnorm(runif(1)*(1-A)+A,lambda[j,1]+lambda[j,2]*z[i],1);
}
}
}#End sample xstar for binary variables
}#End if varcode[j]==3
}

#sample z
sigma2<-1/(1+t(lambda[,2])%*%diag(1/psi)%*%lambda[,2]);
sigma<-sqrt(sigma2);
mu<-sigma2%*%(t(lambda[,2])%*%diag(1/psi)%*%(t(xstar)-lambda[,1]));
z<-rnorm(numrec,mu,sigma*rep(1,length=numrec));

#sample lambda
for(j in 1:numvar){
```

```
#Sample lambda1
if(varcode[j]==1){
lambda[j,1]<-0;
}
else{
sigmaa2<-1/(Lparam[j,1]+numrec/psi[j]);
sigmaa<-sqrt(sigmaa2);
mua<-(Lparam[j,1]*lparam[j,1]+sum(xstar[,j]-z*lambda[j,2])/psi[j])*sigmaa2;
lambda[j,1]<-rnorm(1,mua,sigmaa);
}
#Sample lambda2
sigmab2<-1/(Lparam[j,2]+(z%*%z)/psi[j]);
sigmab<-sqrt(sigmab2);
mub<-(Lparam[j,2]*lparam[j,2]+(z%*%(xstar[,j]-lambda[j,1]))/psi[j])*sigmab2;
ifelse(j==1&templambda2>=0,lambda[j,2],templambda2)
if(signvec[j]==-1){
lambda[j,2]<-qnorm(pnorm(0,mub,sigmab,lower.tail=TRUE)*runif(1),mub,sigmab)
}
else{
if(signvec[j]==1){
lambda[j,2]<-qnorm(pnorm(0,mub,sigmab,lower.tail=FALSE)*runif(1)+
      pnorm(0,mub,sigmab,lower.tail=TRUE),mub,sigmab)
}
else{
lambda[j,2]<-qnorm(runif(1),mub,sigmab)
}
}
}#End sample lambda
lambdavec<-c(lambda[,1],lambda[,2]);
```

```
#Sample psi
for(j in 1:numvar){
shape<-(2*aparam[j]+numrec)/2;
tempvec<-xstar[,j]-lambda[j,1]-lambda[j,2]*z;
scale<-2*bparam[j]+tempvec%*%tempvec;
psi[j]<-ifelse(varcode[j]==1,1/rgamma(1,shape,scale),1);
}

#Colect output
outvec<-c(z,lambdavec,gammavec);
if(n>burnin){
outmat[(n-burnin),]<-outvec;
}
}#End Gibb

outmeanmat<-matrix(0,nrow=(roundnum/batch),ncol=length(outmat.col),
        dimnames=list(c(1:(roundnum/batch)),outmat.col));
for(n in 1:(roundnum/batch)){
outmeanmat[n,]<-outmat[(n*batch),];
}
outmean<-colMeans(outmeanmat);
Means<-outmean[(numrec+1):(numrec+numvar)];
Loadings<-outmean[(numrec+numvar+1):(numrec+(2*numvar))];
Scores<-outmean[1:numrec];
record<-outmat;
lambdaout<-as.data.frame(rbind(Means,Loadings),row.names=c("Means","Loadings"))
names(lambdaout)<-varname;
output<-list(Record=record,Lambda=lambdaout,Scores=Scores);
output
}#End function
```

โปรแกรมสำหรับฟังก์ชัน fmcmcGHR

```
fmcmcGHR<-function(dframe,signvec,roundnum,batch){
levelvec<-sapply(sapply(dframe, levels),length);
varcode<-ifelse(levelvec==0,1,ifelse(levelvec==2,3,2));
numrec<-nrow(dframe);
numvar<-ncol(dframe);
recname<-row.names(dframe);
varname<-names(dframe);
burnin<-10000;

#Standardize continuous data
for(j in 1:numvar){
ifelse(varcode[j]==1,dframe[,j]<-(dframe[,j]-
        mean(dframe[,j]))/sd(dframe[,j]),dframe[,j]<-dframe[,j]);
}

#Prior Parameters
lparam<-matrix(0,numvar,2);
Lparam<-matrix(0.25,numvar,2);
aparam<-rep(0.001/2, length=numvar);
bparam<-rep(0.001/2, length=numvar);
t2param<-rep(0.01, length=numvar);

#Initialize gamma ordmax
gamma<-c();
for(j in 1:numvar){
if(varcode[j]!=2){
subgamma<-list(0);
}
else{
```

```
subgamma<-list(c(0,qnorm((seq(2:(levelvec[j]-1))/(2*levelvec[j])+0.5))));
}
gamma<-c(gamma,subgamma);
}

#Initialize xstar
xstar<-matrix(0,numrec,numvar);
for(j in 1:numvar){
if(varcode[j]==1) xstar[,j]<-dframe[,j];
if(varcode[j]==3) xstar[,j]<-as.numeric(dframe[,j])-1.5;
if(varcode[j]==2){
for(i in 1:numrec){
if(dframe[i,j]==1) xstar[i,j]<-0-0.5;
if((dframe[i,j]!=1)&(dframe[i,j]!=levelvec[j]))xstar[i,j]<-
        (0.5*(gamma[[j]][dframe[i,j]]+gamma[[j]][(as.numeric(dframe[i,j])-1)]));
if(dframe[i,j]==levelvec[j]) xstar[i,j]<-(gamma[[j]][(levelvec[j]-1)]+0.5);
}
}
}

#Initialize z
z<-rnorm(numrec);

#Initialize psi
psi<-rep(0,length=numvar);
psi<-ifelse(varcode==1,aparam*bparam,1);

#Initialize lambda constrain lambda2(courts) to be "-"
lambda<-matrix(0,numvar,2);
```

```
lambda[,1]<-ifelse(varcode==1,0,rnorm(numvar,lparam[,1],1 lambdatemp<-
        qnorm(runif(numvar),0,1);
lambda[,2]<-ifelse(signvec==-1,-
        abs(lambdatemp),ifelse(signvec==1,abs(lambdatemp),lambdatemp));

#Initialize output
lambda.col<-c();
gamma.col<-c();
## Get output column names
for(k in 1:ncol(lambda)){
lambda.col<-c(lambda.col,paste(varname,".l",k,sep=""));
}
for(j in 1:numvar){
if(levelvec[j]>2){
gamma.col<-c(gamma.col,paste(varname[j],".g",1:(levelvec[j]-1),sep=""));
}
}
outmat.col<-c(recname,lambda.col,gamma.col);
outmat<-
        matrix(0,nrow=roundnum,ncol=length(outmat.col),dimnames=list(c(1:roundnum),o
        utmat.col));

###Gibb Sampler
for(n in 1:(roundnum+burnin)){
outvec<-c();
gammavec<-c();
lambdavec<-c();

#Sample gamma
for(j in  1:numvar){
```

```
if(varcode[j]==2){
for(k in 2:(levelvec[j]-1)){
a<-max(max(xstar[,j][dframe[,j]==k]),gamma[[j]][k-1]);
if(k==(levelvec[j]-1)){
b<-min(xstar[,j][dframe[,j]==(k+1)])
}
else{
b<-min(min(xstar[,j][dframe[,j]==(k+1)]),gamma[[j]][k+1]);
}
gamma[[j]][k]<-runif(1,a,b);
}
gammavec<-c(gammavec,gamma[[j]]);
}#End if varcode[j]==2
}#End Sample Gamma[[j]]

#sample xstar for ordinal and binary variables
for(j in 1:numvar){
if(varcode[j]==2){
for(k in 1:levelvec[j]){
x<-xstar[dframe[,j]==k,j];
d<-rnorm(length(x),0,1);
d<-d/sqrt(sum(d^2));
if(k == 1){
gleft<-c();
}
else{
gleft<-rep(gamma[[j]][k-1],length=length(x));
}
if(k==levelvec[j]){
gright<-c();
```

```
}
else{
gright<-rep(gamma[[j]][k],length=length(x));
}
rationvec<-c((gleft-x)/d,(gright-x)/d);
rmin<-max(rationvec[rationvec<0],-Inf);
rmax<-min(rationvec[rationvec>0],Inf);
zx<-z[dframe[,j]==k];
mux<-c(lambda[j,1]+lambda[j,2]*zx);
sigmar2<-1/(d%*%d);
sigmar<-sqrt(sigmar2);
mur<-((mux-x)%*%d)*sigmar2;
A<-pnorm(rmin,mur,sigmar);
B<-pnorm(rmax,mur,sigmar);
r<-qnorm(((B-A)*runif(1))+A,mur,sigmar);
xnew<-x+r*d;
xstar[dframe[,j]==k,j]<-xnew;
}
}
else{
if(varcode[j]==3){
for(i in 1:numrec){
A<-pnorm(0,lambda[j,1]+lambda[j,2]*z[i],1,lower.tail=TRUE);
if(dframe[i,j]==0){
xstar[i,j]<-qnorm(runif(1)*A,lambda[j,1]+lambda[j,2]*z[i],1);
}
else{
xstar[i,j]<-qnorm(runif(1)*(1-A)+A,lambda[j,1]+lambda[j,2]*z[i],1);
}
}
```

```
}#End sample xstar for binary variables
}#End if varcode[j]==3
                    }
#sample z
sigma2<-1/(1+t(lambda[,2])%*%diag(1/psi)%*%lambda[,2]);
sigma<-sqrt(sigma2);
mu<-sigma2%*%(t(lambda[,2])%*%diag(1/psi)%*%(t(xstar)-lambda[,1]));
z<-rnorm(numrec,mu,sigma*rep(1,length=numrec));

#sample lambda
for(j in 1:numvar){
#Sample lambda1
if(varcode[j]==1){
lambda[j,1]<-0;
}
else{
sigmaa2<-1/(Lparam[j,1]+numrec/psi[j]);
sigmaa<-sqrt(sigmaa2);
mua<-(Lparam[j,1]*lparam[j,1]+sum(xstar[,j]-z*lambda[j,2])/psi[j])*sigmaa2;
lambda[j,1]<-rnorm(1,mua,sigmaa);
}
#Sample lambda2
sigmab2<-1/(Lparam[j,2]+(z%*%z)/psi[j]);
sigmab<-sqrt(sigmab2);
mub<-(Lparam[j,2]*lparam[j,2]+(z%*%(xstar[,j]-lambda[j,1]))/psi[j])*sigmab2;
ifelse(j==1&templambda2>=0,lambda[j,2],templambda2)
if(signvec[j]==-1){
lambda[j,2]<-qnorm(pnorm(0,mub,sigmab,lower.tail=TRUE)*runif(1),mub,sigmab)
}
else{
```

```
if(signvec[j]==1){
lambda[j,2]<-qnorm(pnorm(0,mub,sigmab,lower.tail=FALSE)*runif(1)+
        pnorm(0,mub,sigmab,lower.tail=TRUE),mub,sigmab)
}
else{
lambda[j,2]<-qnorm(runif(1),mub,sigmab)
}
}
}#End sample lambda
lambdavec<-c(lambda[,1],lambda[,2]);

#Sample psi
for(j in 1:numvar){
shape<-(2*aparam[j]+numrec)/2;
tempvec<-xstar[,j]-lambda[j,1]-lambda[j,2]*z;
scale<-2*bparam[j]+tempvec%*%tempvec;
psi[j]<-ifelse(varcode[j]==1,1/rgamma(1,shape,scale),1);
}

#Colect output
outvec<-c(z,lambdavec,gammavec);
if(n>burnin){
outmat[(n-burnin),]<-outvec;
}
}#End Gibb

outmeanmat<-matrix(0,nrow=(roundnum/batch),ncol=length(outmat.col),
        dimnames=list(c(1:(roundnum/batch)),outmat.col));
for(n in 1:(roundnum/batch)){
outmeanmat[n,]<-outmat[(n*batch),];
```

```
}
outmean<-colMeans(outmeanmat);
Means<-outmean[(numrec+1):(numrec+numvar)];
Loadings<-outmean[(numrec+numvar+1):(numrec+(2*numvar))];
Scores<-outmean[1:numrec];
record<-outmat;
lambdaout<-as.data.frame(rbind(Means,Loadings),row.names=c("Means","Loadings"))
names(lambdaout)<-varname;
output<-list(Record=record,Lambda=lambdaout,Scores=Scores);
output
}#End function
```

โปรแกรมสำหรับฟังก์ชัน batchMeans

```
batchMeans<-function(x,numbatch){
x<-as.data.frame(x)
roundnum<-dim(x)[1];
batchmat<-{};
size<-roundnum/numbatch;
batchmat<-matrix(0,nrow=numbatch,ncol=length(names(x)),
        dimnames=list(c(1:numbatch),names(x)));

for(i in 1:numbatch){
batch<-colMeans(x[(((i-1)*size)+1):((i*size)),])
batchmat[i,]<-batch;
}
batchmean<-colMeans(batchmat);
batchsd<-apply(batchmat,2,sd);
batchSE<-batchsd/sqrt(numbatch);
data.frame(mean=batchmean,SE=batchSE,sd=batchsd);
}#End function
```

โปรแกรมสำหรับฟังก์ชัน meanplot

```
meanplot<-function(data,range){
numround<-dim(data)[1];
numrec<-dim(data)[2];
meanmat<-matrix(0,nrow=(numround/range),ncol=numrec,
        dimnames=list(c(1:(numround/range)),names(data)));
for(i in 1:(numround/range)){
meanmat[i,]<-colMeans(data[1:(i*range),]);
}
meanmat;
}#End function
```

ภาคผนวก ข

ตารางที่ 1 แสดงคะแนนเฉลี่ย (mean) และอันดับ (Rank) ของแต่ละประเทศจากการสุ่มตัวอย่างแบบฮิตแอนด์รันและการสุ่มตัวอย่างแบบกิบส์

Country	Hit-and-run		Gibbs	
	mean	Rank	mean	Rank
Argentina	0.5157	41	0.4271	39
Australia	-1.1885	9	-1.1647	10
Austria	-1.1077	13	-1.0864	14
Bangladesh	1.4951	60	1.5134	61
Belgium	-1.1100	12	-1.0881	13
Bolivia	1.7553	62	1.7296	62
Botswana	-0.3854	22	-0.3809	22
Brazil	-0.3992	21	-0.3946	21
Burma	0.8431	52	0.8821	52
Cameroon	0.6357	48	0.6746	49
Canada	-1.7241	1	-1.7345	1
Chile	-0.0147	27	-0.0058	28
Colombia	0.3809	37	0.3797	37
Congo-Kinshasa	1.4952	61	1.5132	60
Costa Rica	-0.4431	19	-0.5004	17
Cote d'Ivoire	-0.2236	24	-0.1878	24
Denmark	-1.6779	7	-1.6840	7
Dominican Republic	0.6063	45	0.5729	45
Ecuador	-0.0103	29	-0.0049	29
Finland	-1.6821	5	-1.6903	6
Gambia, The	0.2392	33	0.2717	34
Ghana	0.9643	55	0.9691	53
Greece	0.1407	31	0.1052	31
Hungary	-0.4101	20	-0.4041	20
India	0.2049	32	0.2338	32
Indonesia	0.9540	53	1.0282	54
Iran	1.1417	58	1.0316	55
Ireland	-1.1576	10	-1.1345	11
Israel	-0.2405	23	-0.2768	23

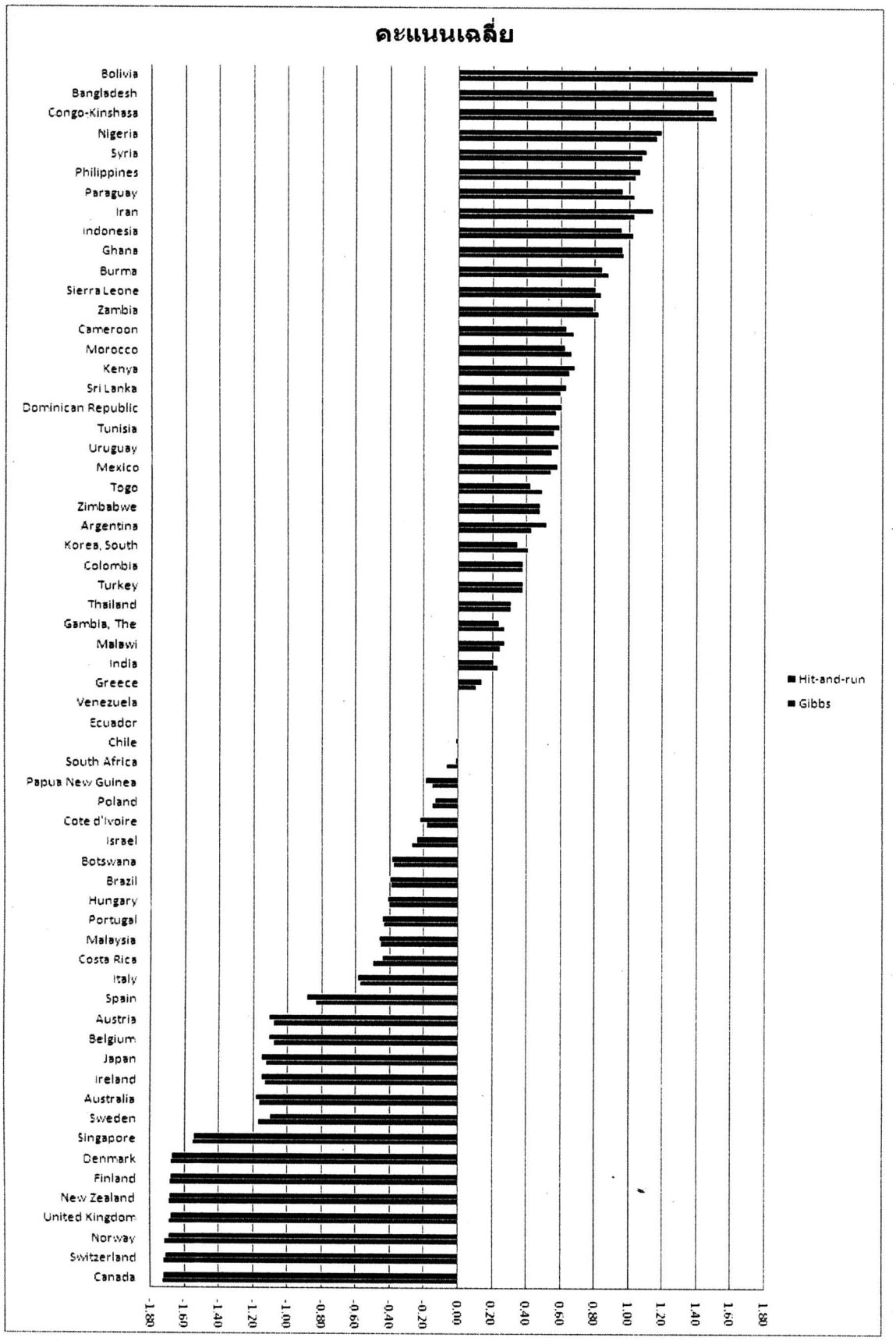
ตารางที่ 1 (ต่อ) แสดงคะแนนเฉลี่ย (mean) และอันดับ (Rank) ของแต่ละประเทศจากการสุ่มตัวอย่างแบบฮิตแอนด์รันและการสุ่มตัวอย่างแบบกิบส์

Country	Hit-and-run		Gibbs	
	mean	Rank	mean	Rank
Italy	-0.5892	16	-0.5768	16
Japan	-1.1573	11	-1.1291	12
Kenya	0.6810	49	0.6495	47
Korea, South	0.3485	36	0.4135	38
Malawi	0.2741	34	0.2435	33
Malaysia	-0.4647	17	-0.4552	18
Mexico	0.5802	42	0.5471	42
Morocco	0.6277	46	0.6614	48
New Zealand	-1.6861	4	-1.6932	5
Nigeria	1.1889	59	1.1651	59
Norway	-1.6925	3	-1.7188	3
Papua New Guinea	-0.1883	25	-0.1509	26
Paraguay	0.9623	54	1.0345	56
Philippines	1.0618	56	1.0375	57
Poland	-0.1330	26	-0.1559	25
Portugal	-0.4470	18	-0.4393	19
Sierra Leone	0.8013	51	0.8330	51
Singapore	-1.5509	8	-1.5574	8
South Africa	-0.0126	28	-0.0733	27
Spain	-0.8881	15	-0.8369	15
Sri Lanka	0.6342	47	0.6029	46
Sweden	-1.1074	14	-1.1719	9
Switzerland	-1.7130	2	-1.7225	2
Syria	1.1005	57	1.0746	58
Thailand	0.3108	35	0.3092	35
Togo	0.4237	39	0.4916	41
Tunisia	0.5926	44	0.5634	44
Turkey	0.3820	38	0.3792	36
United Kingdom	-1.6817	6	-1.6935	4

ตารางที่ 1 (ต่อ) แสดงคะแนนเฉลี่ย (mean) และอันดับ (Rank) ของแต่ละประเทศจากการสุ่มตัวอย่างแบบฮิตแอนด์รันและการสุ่มตัวอย่างแบบกิบส์

Country	Hit-and-run		Gibbs	
	mean	Rank	mean	Rank
Uruguay	0.5860	43	0.5526	43
Venezuela	-0.0041	30	0.0023	30
Zambia	0.7922	50	0.8232	50
Zimbabwe	0.4833	40	0.4801	40

รูปที่ 1 แสดงคะแนนเฉลี่ยเรียงตามความเสี่ยงทางการเมือง-เศรษฐกิจจากน้อยไปหามาก

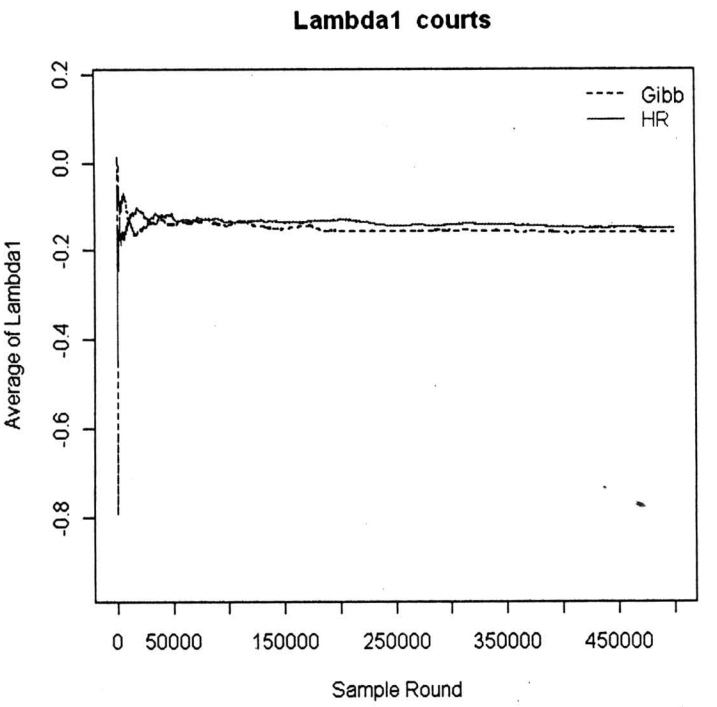


ตารางที่ 2 แสดงค่าประมาณ (mean) และส่วนเบี่ยงเบนมาตรฐาน (SE) ของสัมประสิทธิ์ปัจจัย (factor loading)

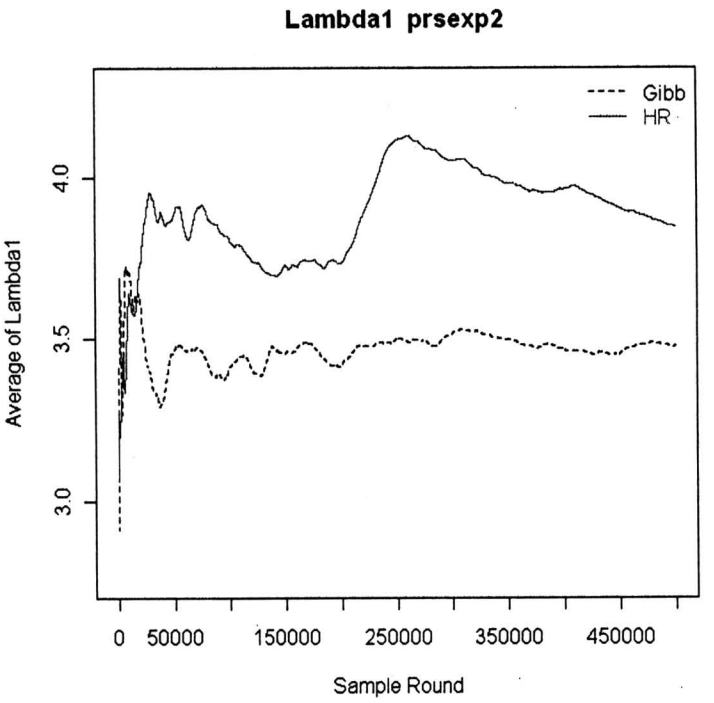
	Hit-and-run		Gibbs	
	mean	SE	mean	SE
Lambda1 courts	-0.1514	0.0067	-0.1596	0.0059*
Lambda1 barb2	0	0	0	0
Lambda1 prsexp2	3.8446	0.1467	3.4764	0.0616*
Lambda1 prscorr2	3.0004	0.0674*	3.2646	0.0866
Lambda1 gdpw2	0	0	0	0
Lambda2 courts	-3.1814	0.0245*	-3.0705	0.0324
Lambda2 barb2	0.7037	0.0009*	0.6997	0.0015
Lambda2 prsexp2	-2.1563	0.0838	-1.9433	0.0374*
Lambda2 prscorr2	-2.1559	0.0534*	-2.3948	0.0718
Lambda2 gdpw2	-0.6682	0.0036	-0.6774	0.0019*

* แสดงว่าส่วนเบี่ยงเบนมาตรฐานจากการสุ่มด้วยวิธีดังกล่าวมีค่าน้อยกว่าการสุ่มตัวอย่างอีกวิธีหนึ่ง

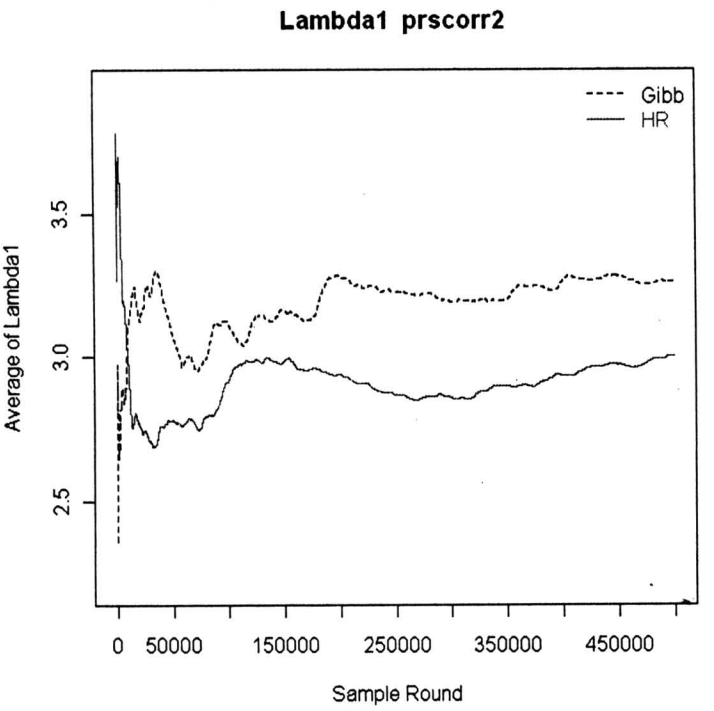
รูปที่ 2 แสดงค่าเฉลี่ยสะสมของ Lambda1 courts



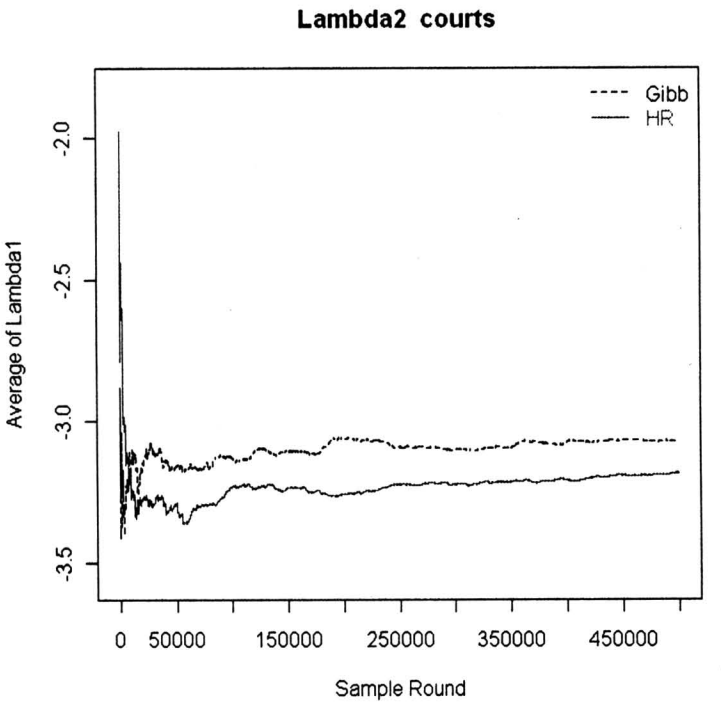
รูปที่ 3 แสดงค่าเฉลี่ยสะสมของ Lambda1 prsexp2



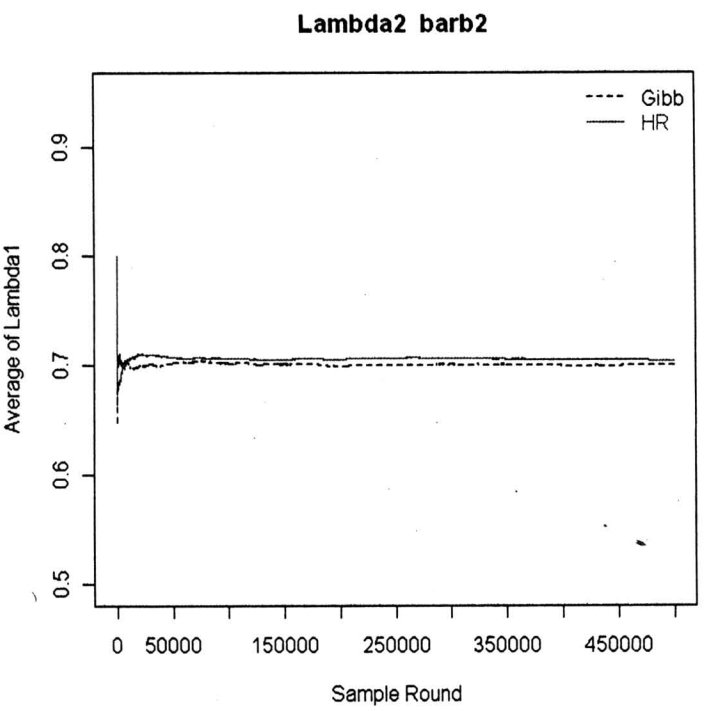
รูปที่ 4 แสดงค่าเฉลี่ยสะสมของ Lambda1 prscorr2



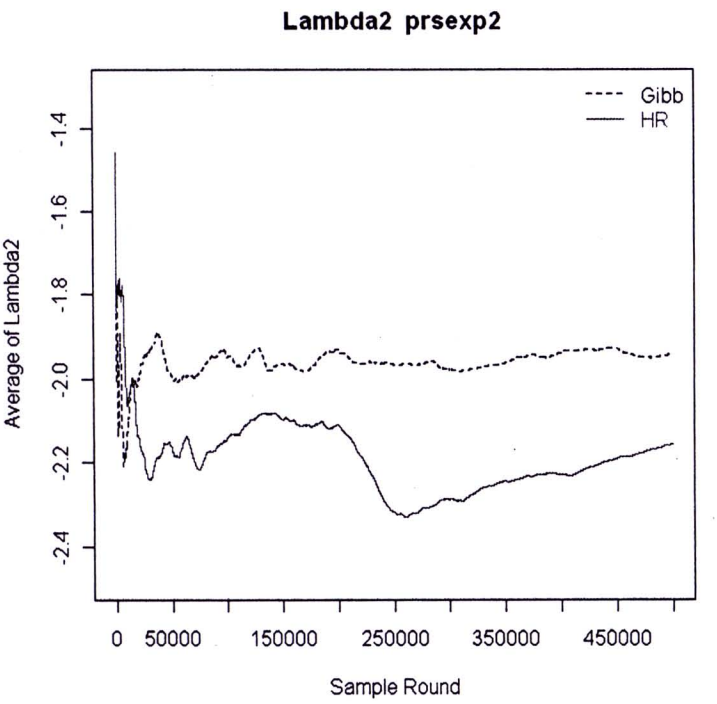
รูปที่ 5 แสดงค่าเฉลี่ยสะสมของ Lambda2 courts



รูปที่ 6 แสดงค่าเฉลี่ยสะสมของ Lambda2 barb2

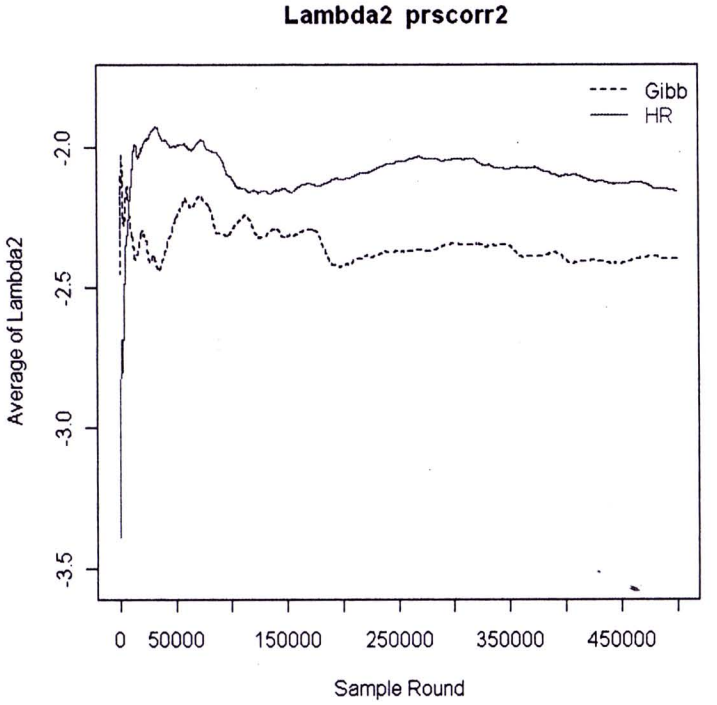


รูปที่ 7 แสดงค่าเฉลี่ยสะสมของ Lambda2 prsexp2



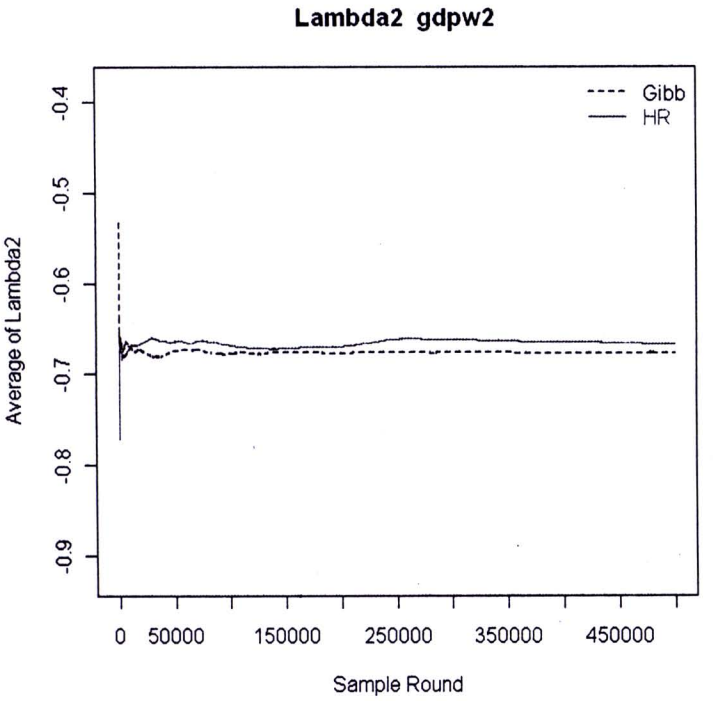
สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

รูปที่ 8 แสดงค่าเฉลี่ยสะสมของ Lambda2 prscorr2



สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

รูปที่ 9 แสดงค่าเฉลี่ยสะสมของ Lambda2 gdpw2

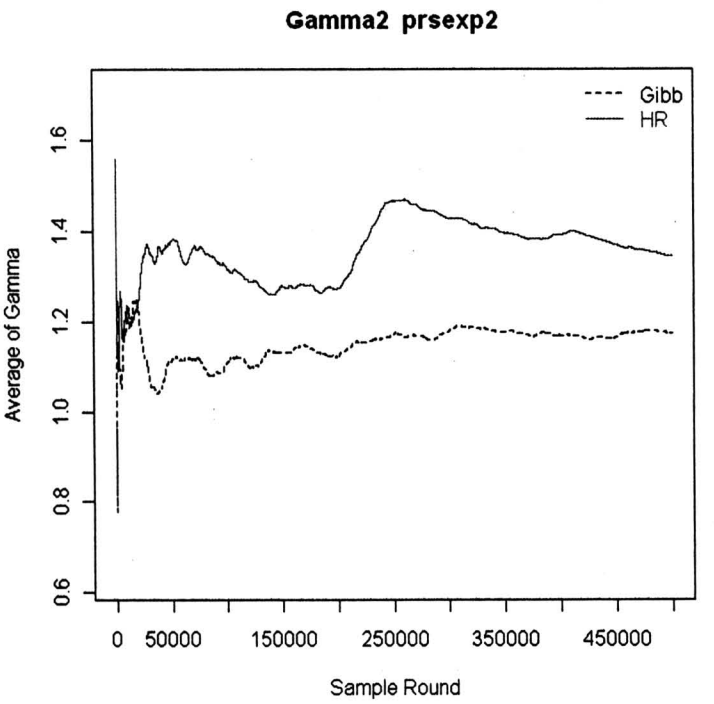


ตารางที่ 3 แสดงค่าประมาณ (mean) และส่วนเบี่ยงเบนมาตรฐาน (SE) ของ Gamma

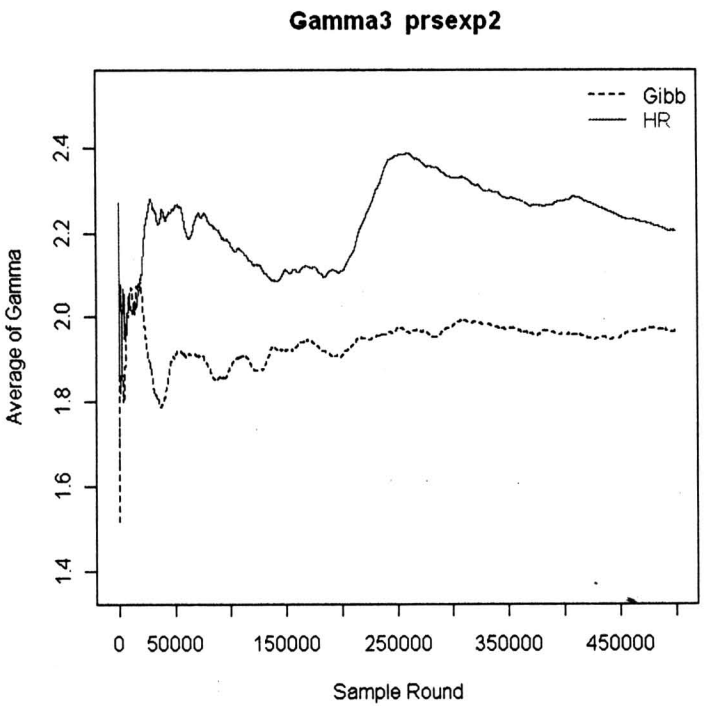
	Hit-and-run		Gibbs	
	mean	SE	mean	SE
Gamma1 prsexp2	0	0	0	0
Gamma2 prsexp2	1.3440	0.0733	1.1747	0.0285*
Gamma3 prsexp2	2.2064	0.1031	1.9662	0.0406*
Gamma4 prsexp2	3.9556	0.1540	3.5670	0.0641*
Gamma5 prsexp2	5.7865	0.2150	5.2483	0.0895*
Gamma1 prscorr2	0	0	0	0
Gamma2 prscorr2	1.4300	0.0353*	1.5483	0.0447
Gamma3 prscorr2	3.1409	0.0725*	3.4157	0.0962
Gamma4 prscorr2	4.4364	0.0965*	4.8289	0.1306
Gamma5 prscorr2	5.8301	0.1264*	6.3776	0.1742

* แสดงว่าส่วนเบี่ยงเบนมาตรฐานจากการสุ่มด้วยวิธีดังกล่าวมีค่าน้อยกว่าการสุ่มตัวอย่างอีกวิธีหนึ่ง

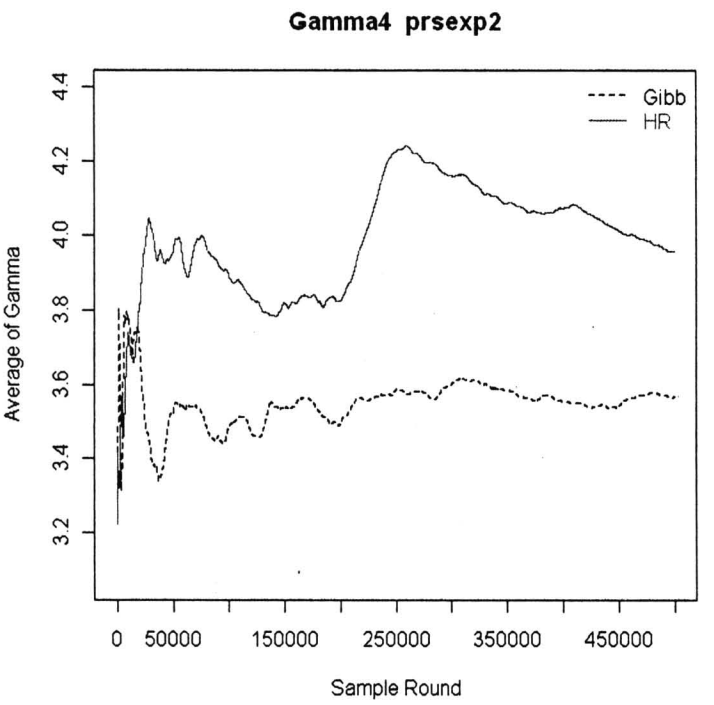
รูปที่ 10 แสดงค่าเฉลี่ยสะสมของ Gamma2 prsexp2



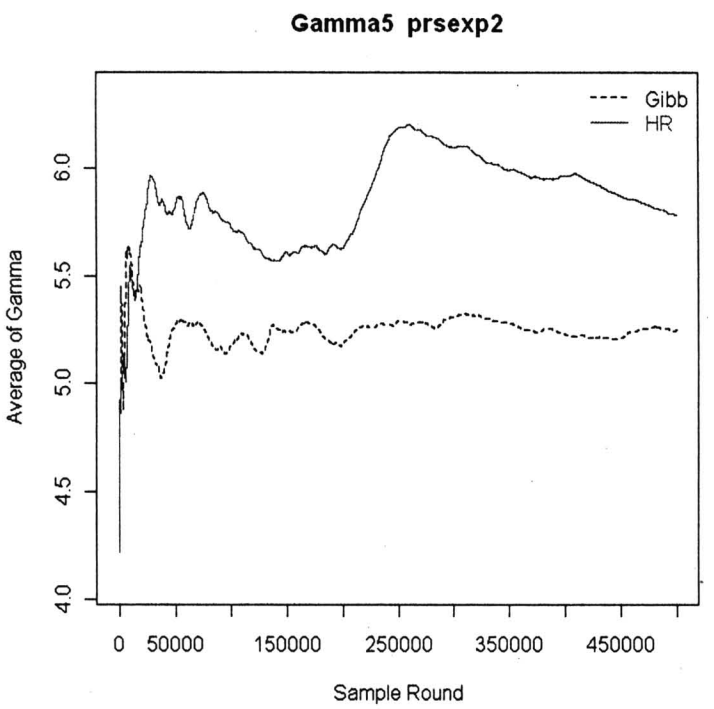
รูปที่ 11 แสดงค่าเฉลี่ยสะสมของ Gamma3 prsexp2



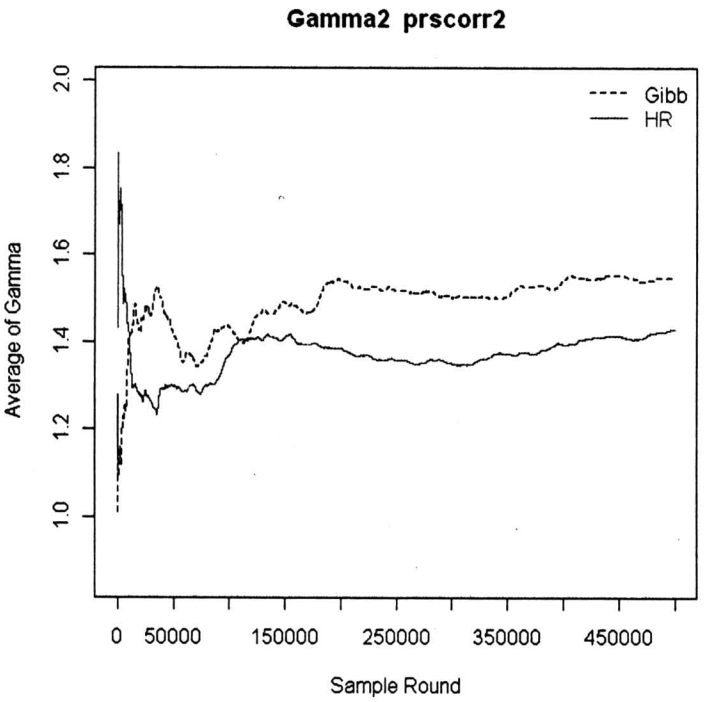
รูปที่ 12 แสดงค่าเฉลี่ยสะสมของ Gamma4 prsexp2



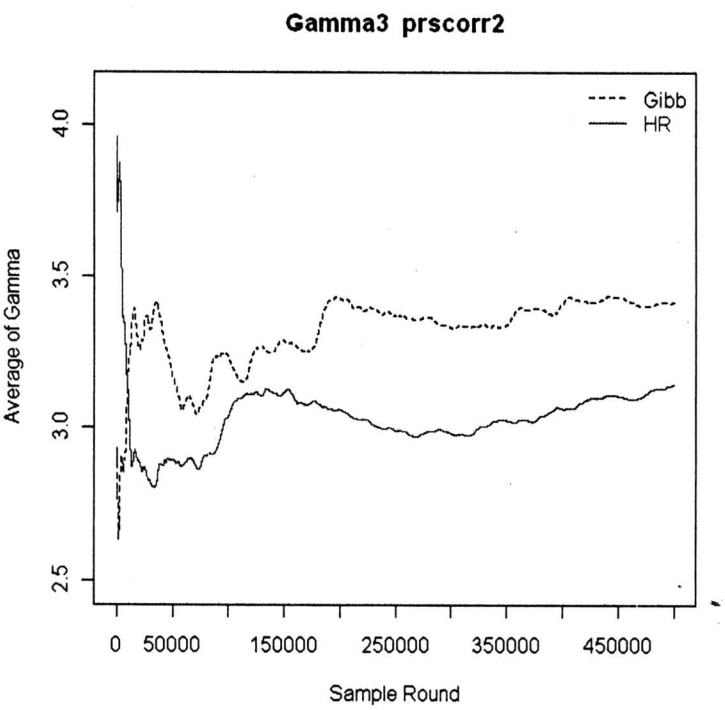
รูปที่ 13 แสดงค่าเฉลี่ยสะสมของ Gamma5 prsexp2



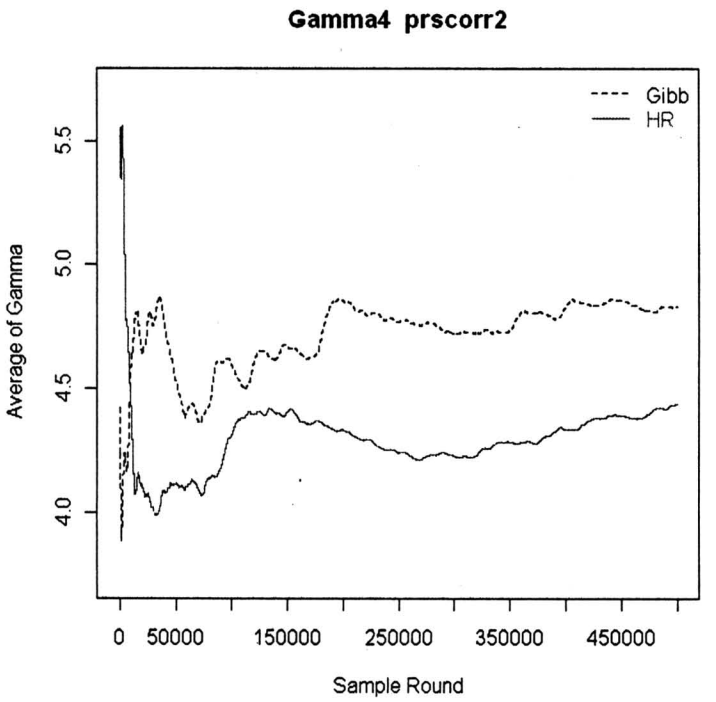
รูปที่ 14 แสดงค่าเฉลี่ยสะสมของ Gamma2 prscorr2



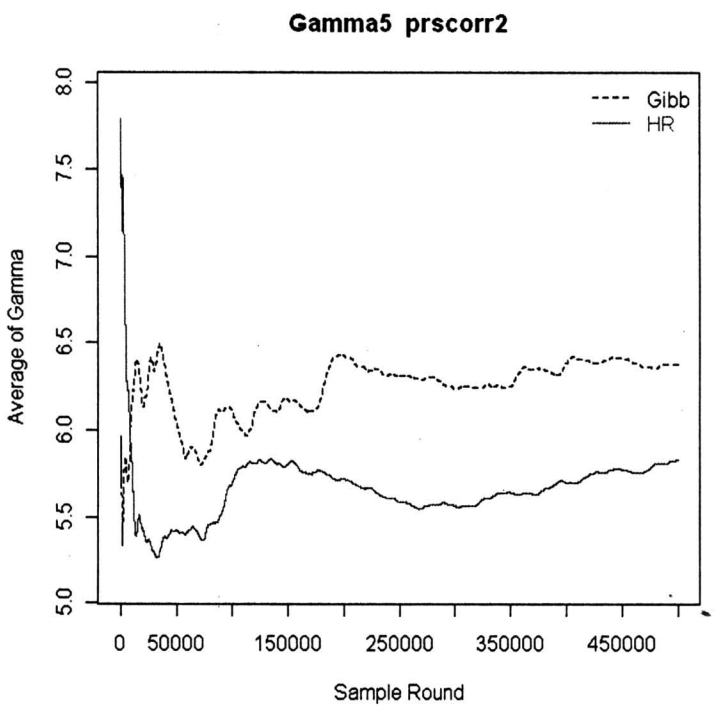
รูปที่ 15 แสดงค่าเฉลี่ยสะสมของ Gamma3 prscorr2



รูปที่ 16 แสดงค่าเฉลี่ยสะสมของ Gamma4 prscorr2



รูปที่ 17 แสดงค่าเฉลี่ยสะสมของ Gamma5 prscorr2



ประวัติผู้เขียนวิทยานิพนธ์

นางสาวไรยา พลเสน เกิดเมื่อวันพุธที่ 16 พฤษภาคม พ.ศ. 2527 สำเร็จการศึกษาระดับปริญญาบัณฑิต หลักสูตรสถิติศาสตรบัณฑิต (สต.บ.) ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2548 และเข้าศึกษาต่อในหลักสูตรสถิติศาสตรมหาบัณฑิต (สต.ม.) สาขาสถิติ ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2551