ภาคผนวก

โปรแกรมที่ใช้ในการวิเคราะห์ข้อมูล

การจำลองข้อมูล

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nu= 50
X1=runif(10000,25,100)
X2=runif(10000,50,100)
X4=runif(10000,87,150)
X5=runif(10000,1,70)
X6=runif(10000,35,90)
X7=runif(10000,3,17)
X8=runif(10000,8,22)
X9=runif(10000,15,80)
X10=runif(10000,20,150)
e=rnorm(10000,0,50)
b0=100, b1=-155,  b2=15 , b3=11.5 , b4=509 , b5=0,  b6=0 ,  b7=4, b8=7,  b9=0, b10=0
samX1=sample(X1,nu,replace=FALSE)   # Sampling
samX2=sample(X2,nu,replace=FALSE)   # with out
samX3= sample(X3,nu,replace=FALSE)
samX4=sample(X4,nu,replace=FALSE)
samX5=sample(X5,nu,replace=FALSE)
samX6=sample(X6,nu,replace=FALSE)
samX7=sample(X7,nu,replace=FALSE)
samX8=sample(X8,nu,replace=FALSE)
samX9=sample(X9,nu,replace=FALSE)
samX10=sample(X10,nu,replace=FALSE)
samE=sample(e,nu,replace=FALSE)
samx1=matrix(samX1,nrow=nu,ncol=1)
samx2=matrix(samX2,nrow=nu,ncol=1)
samx3=matrix(samX3,nrow=nu,ncol=1)
```

```
samx4=matrix(samX4,nrow=nu,ncol=1)
samx5=matrix(samX5,nrow=nu,ncol=1)
samx6=matrix(samX6,nrow=nu,ncol=1)
samx7=matrix(samX7,nrow=nu,ncol=1)
samx8=matrix(samX8,nrow=nu,ncol=1)
samx9=matrix(samX9,nrow=nu,ncol=1)
samx10=matrix(samX10,nrow=nu,ncol=1)
same=matrix(samE,nrow=nu,ncol=1)
samY=b0+(b1*samx1)+(b2*samx2)+(b3*samx3)+(b4*samx4)+(b5*samx5)+(b6*samx6)+(b7*sam
x7)+(b8*samx8)+(b9*samx9)+(b10*samx10)+same
samy=matrix(samY,nrow=nu,ncol=1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

วิธีการคัดเลือกตัวแปร

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
###STEP WISE #####
LinearModel<-
lm(samy~samx1+samx2+samx3+samx4+samx5+samx6+samx7+samx8+samx9+samx10)
s13=(sum((samx1-mean(samx1))*(samx3-mean(samx3))))/(nu-1)
s11=(sum((samx1-mean(samx1))^2))/(nu-1)
s33=(sum((samx3-mean(samx3))^2))/(nu-1)
r13=s13/sqrt(s11*s33)
s23=(sum((samx2-mean(samx1))*(samx3-mean(samx3))))/(nu-1)
s22=(sum((samx2-mean(samx2))^2))/(nu-1)
s33=(sum((samx3-mean(samx3))^2))/(nu-1)
r23=s23/sqrt(s22*s33)
rr=c(r13=r13,r23=r23)
print(rr)
stepwise(LinearModel, direction='forward/backward', criterion='AIC')
############################## DGA ######################################
library(GA)
mod<-lm(samy~samx1+samx2+samx3+samx4+samx5+samx6+samx7+samx8+samx9+samx10)
```

```
x <- model.matrix(mod)[, -1]
y <- model.response(model.frame(mod))
fitness <- function(string) {
 inc <- which(string == 1)
 X <- cbind(1, x[,inc])
 mod <- lm.fit(X, y)
 class(mod) <- "lm"
   -AIC(mod)     }
  GA <- ga("binary", fitness = fitness,popSize=200,pcrossover=0.8,pmutation=0.2, nBits =
ncol(x),names = colnames(x),maxiter = 2000)
mod2 <- lm(samy ~ ., data = data.frame(samy = y, x[,GA@solution [1,] == 1]))
summary(mod2)
OLS<-mod2
X <- model.matrix(OLS)
y <- model.response(model.frame(OLS))
MSE<-function(b,nk1=(nu-11)) {
    if(  ((sum((samy-(X%*%b))^2))/nk1) <0){   -9.9x10^10   }else{
            -((sum((samy-(X%*%b))^2))/nk1)
            }       }
se.coef<-sqrt(diag(vcov(OLS)))
min<-coef(OLS)-3*se.coef
max<-coef(OLS)+3*se.coef
GA <- ga(type = "real-valued", fitness = MSE, min = min, max = max,popSize = 200, pmutation   =
0.2, maxiter = 2000 )
summary(GA)
###################  GA ######################
iter=2000
nx=10
mod<-lm(samy~samx1+samx2+samx3+samx4+samx5+samx6+samx7+samx8+samx9+samx10)
x <- model.matrix(mod)[, -1]
y <- model.response(model.frame(mod))
```

```
for (k in 1:iter){ # begin population string
population <- matrix(NA, nrow =100, ncol=nx)
for (j in 1:nx) {
population[, j] <- round(runif(100))     }
chromosome1=sample(matrix(population,ncol=nx,nrow=1),replace=F)
chromosome2=sample(matrix(population,ncol=nx,nrow=1),replace=F)
parents=rbind(chromosome1,chromosome2)
parents
fitness <- function(parents) {
 inc <- which(parents == 1)
 X <- cbind(1, x[,inc])
mod <- lm.fit(X, y)
class(mod) <- "lm"
AIC(mod)     }
fitpa1<-fitness(parents[1,])
fitpa2<-fitness(parents[2,])
mutation<-         function (parents)
{    mutate <- parents
n <- length(parents)
j <- sample(1:n, size = 1)
mutate[j] <- abs(mutate[j] - 1)
return(mutate) }# Crossover & mutation
 r1=runif(1,0,1)               # Random value  for crossover
if (r1>0.8) {                              # No crossover
r2=runif(1,0,1)   # Random value  for  mutation
if (r2>0.1) {  parents              # No mutation
fitp1<-fitness(parents[1,])
fitp2<-fitness(parents[2,])
if(fitp1<=fitp2){
solution = parents[1,]
fitva[k] =fitp1
```

```
}else{ solution = parents[2,]
fitva[k]=fitp2   }
}else {  muc<-mutation(parents)        # do mutation
fitp1<-fitness(muc[1,])
fitp2<-fitness(muc[2,])
fitp3<-fitness(parents[1,])
fitp4<-fitness(parents[2,])
fit<-c(fitp1,fitp2,fitp3,fitp4)
if(fitp1==min(fit)){
solution = muc[1,]
fitva[k]=fitp1
}else if(fitp2==min(fit)){
solution = muc[2,]
fitva[k]=fitp2
}else if (fitp3==min(fit)) {
solution = parents[1,]
fitva[k]=fitp3
}else if (fitp4==min(fit)){
solution = parents[2,]
fitva[k]=fitp4    } }
}else {  #do crossover
n <- ncol(parents)
children <- matrix(NA, nrow = 2, ncol = n)
crossOverPoint <- sample(0:n, size = 1)
if (crossOverPoint == 0) {         #  crossover  0  point
children[1:2, ] <- parents[2:1, ]
fitc1<-fitness(parents[2,])
fitc2<-fitness(parents[1,])
fit<-c(fitc1,fitc2)
if(fitc1<=fitc2){
solution = parents[2,]
```

```
fitva[k]=fitc1
}else{
solution = parents[1,]
fitva[k]=fitc2  }
}else if (crossOverPoint == n) {     #  crossover  n  point
children <- parents
fitp1<-fitness(parents[1,])
fitp2<-fitness(parents[2,])
if(fitp1<=fitp2){
solution = parents[1,]
fitva[k]=fitp1
}else{
solution = parents[2,]
fitva[k]=fitp2
} }else {        #crossover
children[1, ] <- c(parents[1, 1:crossOverPoint], parents[2, (crossOverPoint + 1):n])
children[2, ] <- c(parents[2, 1:crossOverPoint], parents[1, (crossOverPoint + 1):n])
children<-matrix(c(children[1, ],children[2, ]),ncol=3,nrow=2)
fitp1<-fitness(children[1, ])
fitp2<-fitness(children[2, ])
fitp3<-fitness(parents[1,])
fitp4<-fitness(parents[2,])
fit<-c(fitp1,fitp2,fitp3,fitp4)
if(fitp1==min(fit)){
solution = children[1,]
fitva[k]=fitp1
}else if(fitp2==min(fit)){
solution = children[2,]
fitva[k]=fitp2
}else if (fitp3==min(fit)) {
solution = parents[1,]
```

```
fitva[k]=fitp3
}else if (fitp4==min(fit)){
solution = parents[2,]
fitva[k]=fitp4    } }
r2=runif(1,0,1)  # Do mutation in crossover
children
if (r2>0.1) { children
fitc1<-fitness(children[1,])
fitc2<-fitness(children[2,])
if(fitc1<fitc2){
solution = children[1,]
fitva[k]=fitc1
}else{ solution = children[2,]
fitva[k]=fitc2
}   }else{  muc<-mutation(children)
fitp1<-fitness(muc[1,])
fitp2<-fitness(muc[2,])
fitp3<-fitness(parents[1,])
fitp4<-fitness(parents[2,])
fit<-c(fitp1,fitp2,fitp3,fitp4)
if(fitp1==min(fit)){
solution = muc[1,]
fitva[k]=fitp1
}else if(fitp2==min(fit)){
solution = muc[2,]
fitva[k]=fitp2
}else if (fitp3==min(fit)) {
solution = parents[1,]
fitva[k]=fitp3
}else if (fitp4==min(fit)){
solution = parents[2,]
```

```
fitva[k]=fitp4  }}}

fitva[k]   print(k)   print(fitva[k])

SO=matrix(solution,nrow=1,ncol=nx)

print(SO) }

FIT<-matrix(c(fitva),nrow=1,ncol=iter)

for (ak in 1:iter){

if (min(FIT)==FIT[1,ak]) { print(min(FIT))   print(ak)        break   } }

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

ประวัติผู้เขียน

ชื่อ –นามสกุล	นางสาวศิรินทิพย์ หมื่นจันทร์
วัน เดือน ปี เกิด	16 กันยายน พ.ศ. 2531
ประวัติการศึกษา	ปีการศึกษา 2553 วิทยาศาสตรบัณฑิต สาขาวิชาสถิติ มหาวิทยาลัยนเรศวร
ผลงานทางวิชาการ	นำเสนอผลงานวิจัยเรื่อง “Application of Hybrid Genetic Algorithms for Selection Variable in Multiple Linear Regression Models ” ในการประชุมวิชาการ The 29th National Graduate Research Conference , Mae Fah Luang University, Chiang Rai, Thailand October 24-25, 2013
รางวัลการศึกษา	รางวัลการศึกษายอดเยี่ยมขั้นวิทยาศาสตรบัณฑิต สาขาวิชาสถิติ จากมูลนิธิศาสตราจารย์ ดร.แถบ นีละนิธิ ประจำปี พ.ศ.2554