ภาคผนวก

ผนวก ก

โปรแกรมคอมพิวเตอร์ที่ใช้ในงานวิจัย

การจำลองและการประมวลผลข้อมูลทำได้โดยการใช้โปรแกรม R ในการจำลองตัวแปรอิสระ ตัวแปรตาม ความคลาดเคลื่อนสุ่มตามสถานการณ์ที่ต้องการศึกษา แล้วคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 และกำลังการทดสอบ ตัวอย่างโปรแกรมที่ใช้ประมวลผลเป็นดังนี้

โปรแกรมที่ 1 ตัวอย่างโปรแกรม R สำหรับคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 และกำลังการทดสอบของทั้ง 3 วิธี กรณีแบ่งกลุ่มของข้อมูลโดยใช้ฟังก์ชัน cutree

โปรแกรมที่ 2 ตัวอย่างโปรแกรม R สำหรับคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 และกำลังการทดสอบ ของวิธีเดรปเปอร์และสมิทธิ์ กรณีแบ่งกลุ่มของข้อมูลให้ในแต่ละกลุ่มมีขนาดตัวอย่างเท่าๆ กัน

โปรแกรมที่ 3 ตัวอย่างโปรแกรม R สำหรับคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 และกำลังการทดสอบ ของวิธีซูและยาง กรณีแบ่งกลุ่มของข้อมูลให้ในแต่ละกลุ่มมีขนาดตัวอย่างเท่าๆ กัน

โปรแกรมที่ 4 ฟังก์ชันที่ช่วยในการคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 และกำลังการทดสอบ

โปรแกรมที่ 1 ตัวอย่างโปรแกรม R สำหรับคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 และกำลังการทดสอบของทั้ง 3 วิธี กรณีแบ่งกลุ่มของข้อมูลโดยใช้ฟังก์ชัน cutree

ตัวอย่างโปรแกรมนี้สำหรับสถานการณ์ที่กำหนดให้ $n = 100$ $\beta_0 = 2$ $\beta_1 = 0.0001$, 0.001, 0.01, 0.05, 0.1, 0.5, 1, 1.5 และ 2 เปรียบเทียบ 3 วิธี ได้แก่ วิธีของเดรปเปอร์และสมิทธ์ วิธีของซูและยาง วิธีของมิลเลอร์และนีลล์

```
source("ppo.s")
source("basis.s")
source("CZmat.s")
source("CXmat.s")
source("quantnorep.s")
options(echo=F)
nreject_Drap <- 0
n<-100
stdev<- 1
MIN<-1
MAX<-10
Nsize<- c(2,3,4,5)
ysim<- 2000
ansim<- 10000
alpha<- .05
beta1<- 2
thetaL<- c(0.0001, 0.001, 0.01 , 0.1 ,1,3)

        # thetaL เปลี่ยนไปตามตัวแบบที่ศึกษา

load("x.Rdata")
onesn<- matrix(rep(1,n),byrow=F)
X<- cbind(onesn,x)
load("error.Rdata")
```

```
for (L in 1:length(thetaL)){            #calculate for different theta
theta<-thetaL[L]

#########################################################################
##   simulate for Neill's method
#########################################################################

lof<-beta+(theta*x)
```

ในที่นี้เป็นตัวแบบเส้นตรงจะใช้ในการคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 แต่ในกรณีคำนวณกำลังการทดสอบต้องเปลี่ยนตัวแบบให้เป็นตัวแบบไม่เชิงเส้นซึ่งในการวิจัยนี้มี 5 ตัวแบบคือ ตัวแบบพหุนามลำดับที่ 2 ,3 ตัวแบบตรีโกณมิติแบบที่ 1 ตัวแบบตรีโกณมิติแบบที่ 2 ตัวแบบเลขชี้กำลัง

```
LOF<- matrix(lof,ysim,n,byrow=T)

Y<- LOF+error
xperm<- order(x)

Clist<- lapply(as.list(1:length(Nsize)),function(i,xperm,Nsize)
{Ci<- vector("list",1)
N<- Nsize[i]
Blk<- floor(length(xperm)/N)
start1<- 1
if(1<=Blk)
{for(j in 1:Blk)
{mj<- ((j-1)*N)+start1
Ci[[length(Ci)+1]]<- xperm[mj:(mj+N-1)]}
}
resid<- length(xperm)-(Blk*N)
if(resid>0)
```

```
#exclude singletons if(resid>1)
{start2<- (Blk*N)+start1
Ci[[length(Ci)+1]]<- xperm[start2:length(xperm)]}
Ci[[1]]<- NULL
return(Ci)
},xperm,Nsize)
NC<- length(Clist)
Cmatlist<- lapply(Clist,function(C,n)
{Cmat<- CZmat(C,n)
return(Cmat)
},n)
MBlist<- lapply(Cmatlist,function(Cmat,X)
{MB<- ppo(Cmat)-ppo(ppo(Cmat)%*%X)
return(MB)
},X)
dfBlist<- lapply(MBlist,function(MB)
{dfB<- basis(MB)
return(dfB)
})
MXperp<- diag(n)-ppo(X)
dimCXperp<- basis(MXperp)
MWSlist<- lapply(MBlist,function(MB,MXperp)
{MWS<- MXperp-MB
return(MWS)
},MXperp)
dfWSlist<- lapply(as.list(1:NC),function(j,dfBlist,dimCXperp)
{dfWS<- dimCXperp - dfBlist[[j]]
return(dfWS)
},dfBlist,dimCXperp)
```

```
an<- quantnorep(n,ansim,NC,alpha,MBlist,MWSlist,dfBlist,dfWSlist)
Tvals<- apply(Y,1,function(y,NC,MBlist,MWSlist,dfBlist,dfWSlist,an)
{
bestcomp<- lapply(as.list(1:NC),function(j,y,an,MBlist,MWSlist,dfBlist,dfWSlist)
{
MSB<- (sum((MBlist[[j]]%*%y)^2))/dfBlist[[j]]
MSWS<- (sum((MWSlist[[j]]%*%y)^2))/dfWSlist[[j]]

FB<- MSB/MSWS
FWS<- 1/FB

Bcpt<- qf(1-an,dfBlist[[j]],dfWSlist[[j]])
WScpt<- qf(1-an,dfWSlist[[j]],dfBlist[[j]])
diffB<- FB-as.numeric(Bcpt)
diffWS<- FWS-as.numeric(WScpt)
best<- max(diffB,diffWS)
return(best)
},y,an,MBlist,MWSlist,dfBlist,dfWSlist)

Tval<- max(unlist(bestcomp))
return(Tval)
},NC,MBlist,MWSlist,dfBlist,dfWSlist,an)
nreject<- Tvals > 0
simpower<- mean(nreject)

cat("T simulated power based on Neill's method \n")
print(simpower)
#Tstats<- summary(Tvals)
#cat("T statistic summary\n")
#print(Tstats)
```

```
###################################################################
## simulate for yang's overall test
###################################################################

cat(" Simpower based on Yang's test is \n")
cluster_n<-c(3,4,5,7,10,13,17,20,25,33,50)
```

จำนวนกลุ่มที่ใช้สำหรับวิธีซูและยางเมื่อ n =100 ซึ่งต้องเปลี่ยนไปตามขนาดตัวอย่าง

```
x1<-matrix(sort(x),ncol=1)
linear_md<- beta+(theta*x1)
```

ในที่นี้เป็นตัวแบบเส้นตรงจะใช้ในการคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 แต่ในกรณีคำนวณกำลังการทดสอบต้องเปลี่ยนตัวแบบให้เป็นตัวแบบไม่เชิงเส้นซึ่งในการวิจัยนี้มี 5 ตัวแบบ คือ ตัวแบบพหุนามลำดับที่ 2 ,3 ตัวแบบตรีโกณมิติแบบที่ 1 ตัวแบบตรีโกณมิติแบบที่ 2 ตัวแบบเลขชี้กำลัง

```
for (cl in 1:length(cluster_n)){
Blk<-cluster_n[cl]
cluster<-cutree(hclust(dist(x1),method="complete"),Blk)
```

การแบ่งข้อมูลเป็นกลุ่มโดยใช้ฟังก์ชัน cutree

```
xperm_yang<-rep(1:n,1)
Blksizelist<-vector("list",1)
for (i in 1:Blk) {
Blksizelist[length(Blksizelist)+1]<-sum(cluster==i)}
Blksizelist[1]<-NULL
Blksize<-unlist(Blksizelist)

C_yang<- vector("list",1)
N<-0
for (i in 1:length(Blksize)){
if(i==1) {N=Blksize[i]}
if(i>1){N=N+Blksize[i]}
```

```
start<-N-Blksize[i]+1
C_yang[[length(C_yang)+1]]<-xperm_yang[start:N]
}
C_yang[1]<- NULL
Cmat_yang<- CZmat(C_yang,n)
Xmat_yang<-CXmat(C_yang,x1,n)
X_yang<-cbind(onesn,x1)

########### Get XW matrix ###########

XW0<-lapply(as.list(1:length(Blksize)),function(i,Cmat_yang,Xmat_yang)
{
C1<-matrix(Cmat_yang[,i],ncol=1)
X1<-matrix(Xmat_yang[,i],ncol=1)
X12<-X1**2
Wi<-cbind(C1,X1,X12)
return(Wi)},Cmat_yang,Xmat_yang)
XW1<-matrix(unlist(XW0),n, 3*length(Blksize))
XW<-cbind(X_yang,XW1)

ytrue<- linear_md

YTRUE<-matrix(ytrue,n,byrow=T)
nreject_yang<-matrix(100)
for (k in 1:ysim){
error_yang<- matrix(unlist(error[k,]),nrow=n)
Y_yang<- YTRUE+error_yang

########### get ssex and dfssex ###########

dfssex<-basis( diag(n)-ppo(X_yang))
ssex<-t(Y_yang)%*%(diag(n)-ppo(X_yang))%*%Y_yang
```

```
dfssexw<-basis(diag(n)-ppo(XW))
ssexw<-t(Y_yang)%*%(diag(n)-ppo(XW))%*%Y_yang
ssnum<-ssex-ssexw
dfnum<-dfssex-dfssexw
msnum<-ssnum/dfnum
ssden<-ssexw
dfden<-dfssexw
msden<-ssden/dfden
F<-msnum/msden
Fc<-qf(1-alpha,dfnum,dfden)
nreject_yang[k]<-F>Fc
}
###########   get test power ############
simpower_yang<-mean(nreject_yang)
print(simpower_yang)
}

########################################################################
###simulate for Drepper test
########################################################################

cat(" Simpower based on Draper's test is \n")
cluster_n<-c(3,4,5,7,10,13,17,20,25,33,50)

            # การแบ่งข้อมูลเป็นกลุ่มโดยใช้ฟังก์ชัน cutree

x1<-matrix(sort(x),ncol=1)
for (cl in 1:length(cluster_n)){
Block<-cluster_n[cl]
cluster<-cutree(hclust(dist(x1),method="complete"),Block)
xperm_Drap<-rep(1:n,1)
Blocksizelist<-vector("list",1)
```

```
for (i in 1:Block) {
Blocksizelist[length(Blocksizelist)+1]<-sum(cluster==i)}
Blocksizelist[1]<-NULL
Blocksize<-unlist(Blocksizelist)
C_Drap<- vector("list",1)
C_Drap2<- vector("list",1)
C_Drap3<- vector("list",1)
mean_repx <- vector("list",1)
N<-0
for (i in 1:length(Blocksize)){
if(i==1) {N=Blocksize[i]}
if(i>1){N=N+Blocksize[i]}
start<-N-Blocksize[i]+1
C_Drap[[length(C_Drap)+1]]<-xperm_Drap[start:N]
C_Drap2[[length(C_Drap2)+1]]<-x1[start:N]
C_Drap3[[length(C_Drap3)+1]]<-mean(x1[start:N])
mean_repx[[length(mean_repx)+1]] <- rep(C_Drap3[[i+1]], length(C_Drap2[[i+1]]))
}
C_Drap[1]<- NULL
C_Drap2[1]<- NULL
C_Drap3[1]<- NULL
mean_repx[1] <- NULL
tranx <- unlist(mean_repx)
for (k in 1:ysim){
error_Drap <- unlist(error[k,])
y <- beta+(theta*(tranx)) + error_Drap
```

ในที่นี้เป็นตัวแบบเส้นตรงจะใช้ในการคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 แต่ในกรณีคำนวณกำลังการทดสอบต้องเปลี่ยนตัวแบบให้เป็นตัวแบบไม่เชิงเส้นซึ่งในการวิจัยนี้มี 5 ตัวแบบ คือ ตัวแบบพหุนามลำดับที่ 2 ,3 ตัวแบบตรีโกณมิติแบบที่ 1 ตัวแบบตรีโกณมิติแบบที่ 2 ตัวแบบเลขชี้กำลัง

```
############# fitted model ######################

          fitted_model <- lm(y~tranx)
          sse <- anova(fitted_model)$Sum[2]
          dfsse <- anova(fitted_model)$Df[2]

###### One-way analysis of variance ########

          aov <- aov(y ~ as.factor(tranx))
          sspe <- anova(aov)$Sum[2]
          dfpe <- anova(aov)$Df[2]

########## lack of fittness ###############

          sslof<- sse - sspe
          dflof <- dfsse - dfpe
          F_lof <- (sslof/dflof)/(sspe/dfpe)

          Pval_lof  <- pf(F_lof, dflof, dfpe, lower.tail=F)

          nreject_Drap[k] <- Pval_lof < 0.05
}
          simpower <- mean(nreject_Drap)
          print(simpower)
}
}
```

โปรแกรมที่ 2 ตัวอย่างโปรแกรม R สำหรับคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 และกำลังการทดสอบ ของวิธีเดรปเปอร์และสมิทธิ์ กรณีแบ่งกลุ่มของข้อมูลให้ในแต่ละกลุ่มให้มีขนาดตัวอย่างเท่าๆ กัน

ตัวอย่างโปรแกรมนี้สำหรับสถานการณ์ที่กำหนดให้ $n = 100$ $\beta_0 = 2$ $\beta_1 = 0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 1, 1.5$ และ 2 วิธีเดรปเปอร์และสมิทธิ์ กรณีแบ่งกลุ่มของข้อมูลในแต่ละกลุ่มให้มีขนาดตัวอย่างเท่าๆ กัน

```
source("ppo.s")
source("basis.s")
source("CZmat.s")
source("CXmat.s")
source("quantnorep.s")
options(echo=F)
nreject_Drap <- 0
n<-100
stdev<- 1
ysim<- 2000
alpha<- .05
thetaL<- c(0.0001, 0.001, 0.01 , 0.1 ,1,3)

          # thetaL เปลี่ยนไปตามตัวแบบที่ศึกษา

load("x.Rdata")
load("error.Rdata")
for (L in 1:length(thetaL)){ #calculate for different theta
theta<-thetaL[L]
```

```
########################################################################
##### simulate for Draper test
########################################################################
cat(" Simpower based on Draper's test is \n")
Blocksize<-c(25,25,25,25)
                # แบ่งข้อมูลในแต่ละกลุ่มให้มีขนาดตัวอย่างเท่าๆ กัน  ในที่นี้แบ่งกลุ่มให้มี4 กลุ่ม
ขนาดตัวอย่างกลุ่มละ 25
x1<-matrix(sort(x),ncol=1)
xperm_Drap<-rep(1:n,1)
C_Drap<- vector("list",1)
C_Drap2<- vector("list",1)
C_Drap3<- vector("list",1)
mean_repx <- vector("list",1)
N<-0
for (i in 1:length(Blocksize)){
if(i==1) {N=Blocksize[i]}
if(i>1){N=N+Blocksize[i]}
start<-N-Blocksize[i]+1
C_Drap[[length(C_Drap)+1]]<-xperm_Drap[start:N]
C_Drap2[[length(C_Drap2)+1]]<-x1[start:N]
C_Drap3[[length(C_Drap3)+1]]<-mean(x1[start:N])
mean_repx[[length(mean_repx)+1]] <- rep(C_Drap3[[i+1]], length(C_Drap2[[i+1]]))
}
C_Drap[1]<- NULL
C_Drap2[1]<- NULL
C_Drap3[1]<- NULL
mean_repx[1] <- NULL
tranx <- unlist(mean_repx)
for (k in 1:ysim){
error_Drap <- unlist(error[k,])
```

```
y <-beta+(theta*tranx)+error_Drap
```

ในที่นี้เป็นตัวแบบเส้นตรงจะใช้ในการคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 แต่ในกรณีคำนวณกำลังการทดสอบต้องเปลี่ยนตัวแบบให้เป็นตัวแบบไม่เชิงเส้นซึ่งในการวิจัยนี้มี 5 ตัวแบบ คือ ตัวแบบพหุนามลำดับที่ 2 ,3 ตัวแบบตรีโกณมิติแบบที่ 1 ตัวแบบตรีโกณมิติแบบที่ 2 ตัวแบบเลขชี้กำลัง

```
################## fitted model #######################

        fitted_model <- lm(y~tranx)
        sse <- anova(fitted_model)$Sum[2]
        dfsse <- anova(fitted_model)$Df[2]

########### One-way analysis of variance #########

        aov <- aov(y ~ as.factor(tranx))
        sspe <- anova(aov)$Sum[2]
        dfpe <- anova(aov)$Df[2]

################# lack of fittness ##################
        sslof<- sse - sspe
        dflof <- dfsse - dfpe
        F_lof <- (sslof/dflof)/(sspe/dfpe)
        Pval_lof  <- pf(F_lof, dflof, dfpe, lower.tail=F)
        nreject_Drap[k] <- Pval_lof < 0.05
}
        simpower <- mean(nreject_Drap)
        print(simpower)
}
```

โปรแกรมที่ 3 ตัวอย่างโปรแกรม R สำหรับคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 และกำลังการทดสอบ ของวิธีซูและยาง กรณีแบ่งกลุ่มของข้อมูลให้ในแต่ละกลุ่มมีขนาดตัวอย่างเท่า ๆ กัน

ตัวอย่างโปรแกรมนี้สำหรับสถานการณ์ที่กำหนดให้ $n = 100$ $\beta_0 = 2$ $\beta_1 = 0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 1, 1.5$ และ 2 วิธีซูและยาง กรณีแบ่งกลุ่มของข้อมูลให้ในแต่ละกลุ่มมีขนาดตัวอย่างเท่าๆ กัน

```
source("ppo.s")
source("basis.s")
source("CZmat.s")
source("CXmat.s")
source("quantnorep.s")
options(echo=F)
n<-100
stdev<- 1
ysim<- 2000
alpha<- .05
thetaL<- c(0.0001, 0.001, 0.01 , 0.1 ,1,3)

          # thetaL เปลี่ยนไปตามตัวแบบที่ศึกษา

load("x.Rdata")

onesn<-matrix(rep(1,n),byrow=F)
X<-cbind(onesn,x)

load("error.Rdata")

for (L in 1:length(thetaL)){ #calculate for different theta

theta<-thetaL[L]
```

```
########################################################################
## simulate for yang's overall test
########################################################################
cat(" Simpower based on Yang's test is \n")
Blksize<-c(25,25,25,25)
```

แบ่งข้อมูลในแต่ละกลุ่มให้มีขนาดตัวอย่างเท่าๆ กัน ในที่นี้แบ่งกลุ่มให้มี4 กลุ่ม ขนาดตัวอย่างกลุ่มละ 25

```
x1<-matrix(sort(x),ncol=1)
linear_rg <-beta+(theta*x1)
```

ในที่นี้เป็นตัวแบบเส้นตรงจะใช้ในการคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 แต่ในกรณีคำนวณกำลังการทดสอบต้องเปลี่ยนตัวแบบให้เป็นตัวแบบไม่เชิงเส้นซึ่งในการวิจัยนี้มี 5 ตัวแบบ คือ ตัวแบบพหุนามลำดับที่ 2 ,3 ตัวแบบตรีโกณมิติแบบที่ 1 ตัวแบบตรีโกณมิติแบบที่ 2 ตัวแบบเลขชี้กำลัง

```
xperm_yang<-rep(1:n,1)
C_yang<- vector("list",1)
N<-0
for (i in 1:length(Blksize)){
if(i==1) {N=Blksize[i]}
if(i>1){N=N+Blksize[i]}
start<-N-Blksize[i]+1
C_yang[[length(C_yang)+1]]<-xperm_yang[start:N]
}
C_yang[1]<- NULL
Cmat_yang<- CZmat(C_yang,n)
Xmat_yang<-CXmat(C_yang,x1,n)
X_yang<-cbind(onesn,x1)
```

```
############ Get XW matrix ##############
XW0<-lapply(as.list(1:length(Blksize)),function(i,Cmat_yang,Xmat_yang)
{
C1<-matrix(Cmat_yang[,i],ncol=1)
X1<-matrix(Xmat_yang[,i],ncol=1)
X12<-X1**2
Wi<-cbind(C1,X1,X12)
return(Wi)},Cmat_yang,Xmat_yang)
XW1<-matrix(unlist(XW0),n, 3*length(Blksize))
XW<-cbind(X_yang,XW1)

ytrue<-linear_rg

YTRUE<-matrix(ytrue,n,byrow=T)
nreject_yang<-matrix(100)
for (k in 1:ysim){
error_yang<- matrix(unlist(error[k,]),nrow=n)


###########  extend #############
error_begin<-t(error)
combind<-cbind(x,error_begin)
rank_by_x<-combind[order(x),]
mat_error<-rank_by_x[,-1]
error_rank<-t(mat_error)

############  finish ##############

error_yang<- matrix(unlist(error_rank[k,]),nrow=n)
Y_yang<- YTRUE+error_yang
```

```
##############   get ssex and dfssex #############

dfssex<-basis( diag(n)-ppo(X_yang))
ssex<-t(Y_yang)%*%(diag(n)-ppo(X_yang))%*%Y_yang
dfssexw<-basis(diag(n)-ppo(XW))
ssexw<-t(Y_yang)%*%(diag(n)-ppo(XW))%*%Y_yang
ssnum<-ssex-ssexw
dfnum<-dfssex-dfssexw
msnum<-ssnum/dfnum
ssden<-ssexw
dfden<-dfssexw
msden<-ssden/dfden
F<-msnum/msden
Fc<-qf(1-alpha,dfnum,dfden)
nreject_yang[k]<-F>Fc
}
simpower_yang<-mean(nreject_yang)
print(simpower_yang)
}
```

โปรแกรมที่ 4 ฟังก์ชันที่ช่วยในการคำนวณค่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนประเภทที่ 1 และกำลังการทดสอบ

function CXmat

```
CXmat<- function(cl,x,n)
        {CX<- matrix(rep(0,n*length(cl)),ncol=length(cl))

        for(j in 1:length(cl))
                {cverts<- matrix(rep(0,n),ncol=1)
                vertj<- cl[[j]]
                    for(k in 1:length(vertj))
                        {cverts[vertj[k],]<- x1[vertj[k]]}
                CX[,j]<- cverts}
        return(CX)}
```

function CZmat

```
CZmat<- function(cl,n)
      {CZ<- matrix(rep(0,n*length(cl)),ncol=length(cl))

      for(j in 1:length(cl))
        {cverts<- matrix(rep(0,n),ncol=1)
         vertj<- cl[[j]]
            for(k in 1:length(vertj))
                {cverts[vertj[k],]<- 1}
         CZ[,j]<- cverts}
      return(CZ)}
```

function basis

```
basis<- function(A)
{
B<- A%*%t(A)
m<- nrow(B)
e<- eigen(B,symmetric=T)
vals<- e$values
vals<- unlist(vals)
#cat("eigenvalues\n")
#print(vals)
vecs<- e$vectors
nzvals<- vals[abs(vals)>1.0e-6]
rankA<- length(nzvals)
#basis<- vecs[1:m,1:k]
#return(basis)
return(rankA)
}
```

function ppo

```
require(MASS)
ppo<- function(C)
{
M<- C%*%ginv(t(C)%*%C)%*%t(C)
return(M)
}
```

function quantnorep

```
quantnorep<- function(n,ansim,NC,alpha,Mnumlist,Mdenlist,dfnumlist,dfdenlist)
{
     error<- matrix(rnorm(n*ansim,mean=0,sd=1),ansim,n,byrow=T)
     infs<- apply(error,1,function(e,NC,Mnumlist,Mdenlist,dfnumlist,dfdenlist)
           {
    Fpvals<-lapply(as.list(1:NC),function(j,e,Mnumlist,Mdenlist,dfnumlist,dfdenlist)
           {
           MSnum<- (sum((Mnumlist[[j]]%*%e)^2))/dfnumlist[[j]]
           MSden<- (sum((Mdenlist[[j]]%*%e)^2))/dfdenlist[[j]]
           F<- MSnum/MSden
           Fpval<- 1-pf(F,dfnumlist[[j]],dfdenlist[[j]])
           return(Fpval)
                 },e,Mnumlist,Mdenlist,dfnumlist,dfdenlist)
Fcompvals<- 1-unlist(Fpvals)
pvals<- c(unlist(Fpvals),Fcompvals)
inf<- min(pvals)
return(inf)
       },NC,Mnumlist,Mdenlist,dfnumlist,dfdenlist)
an<- quantile(infs,alpha)
return(an)
}
```

ผนวก ข

กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y

รูปที่ 1 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบเส้นตรง

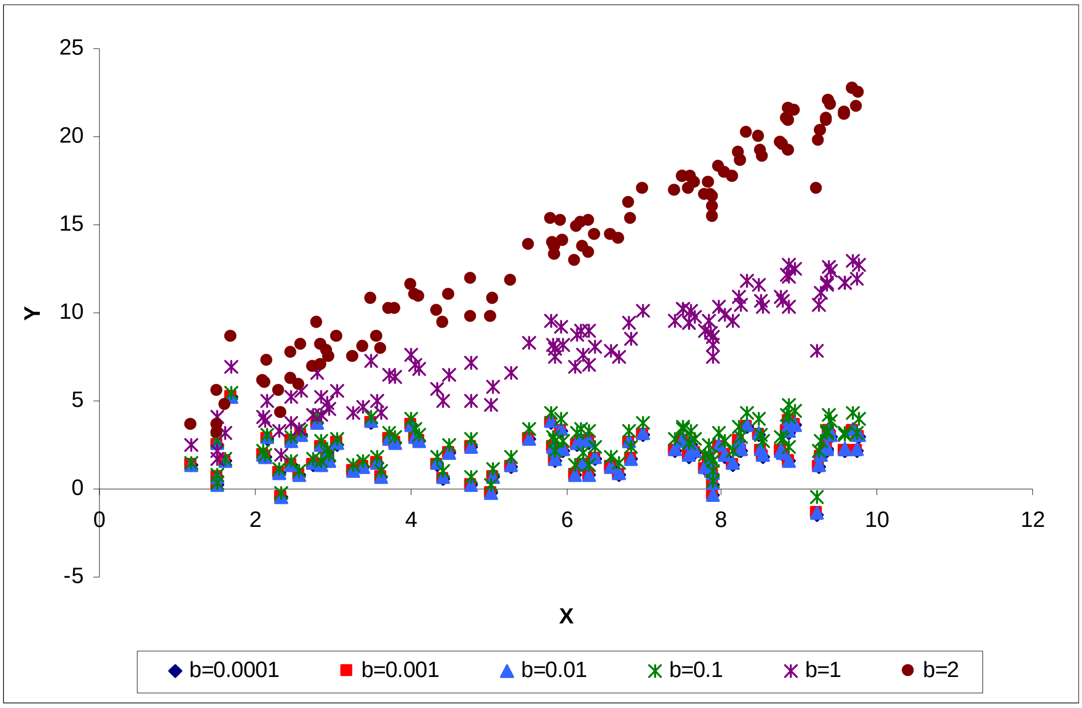
รูปที่ 2 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบพหุนามลำดับที่ 2

รูปที่ 3 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบพหุนามลำดับที่ 3

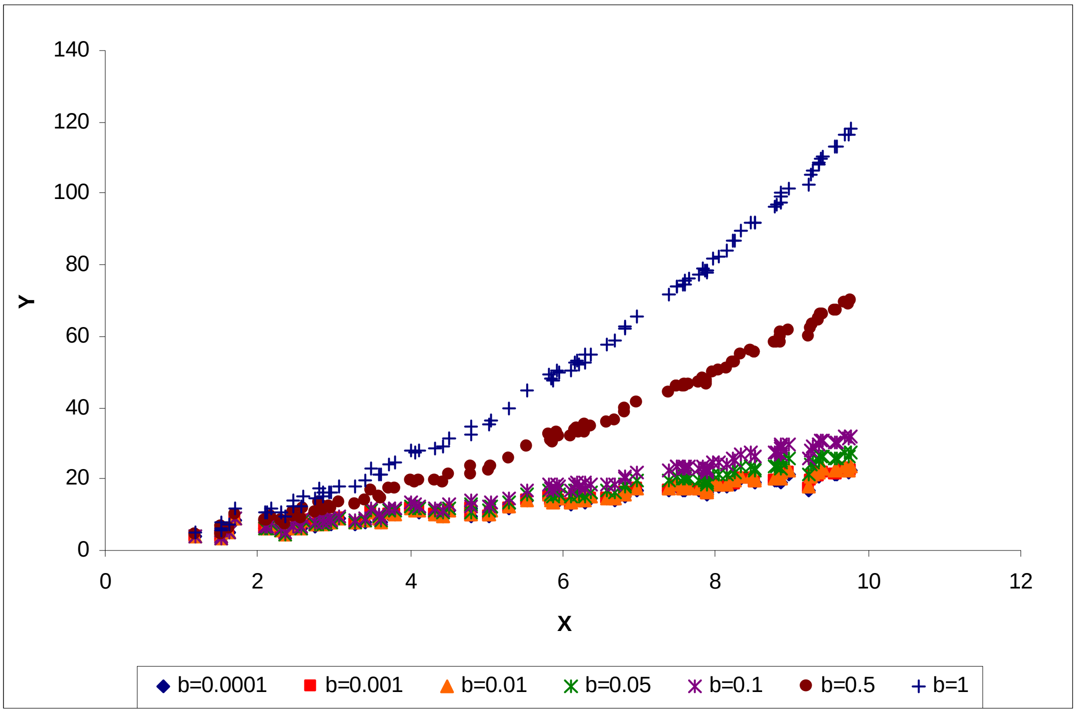
รูปที่ 4 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบฟังชันก์ไซน์

รูปที่ 5 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบตรีโกณมิติ

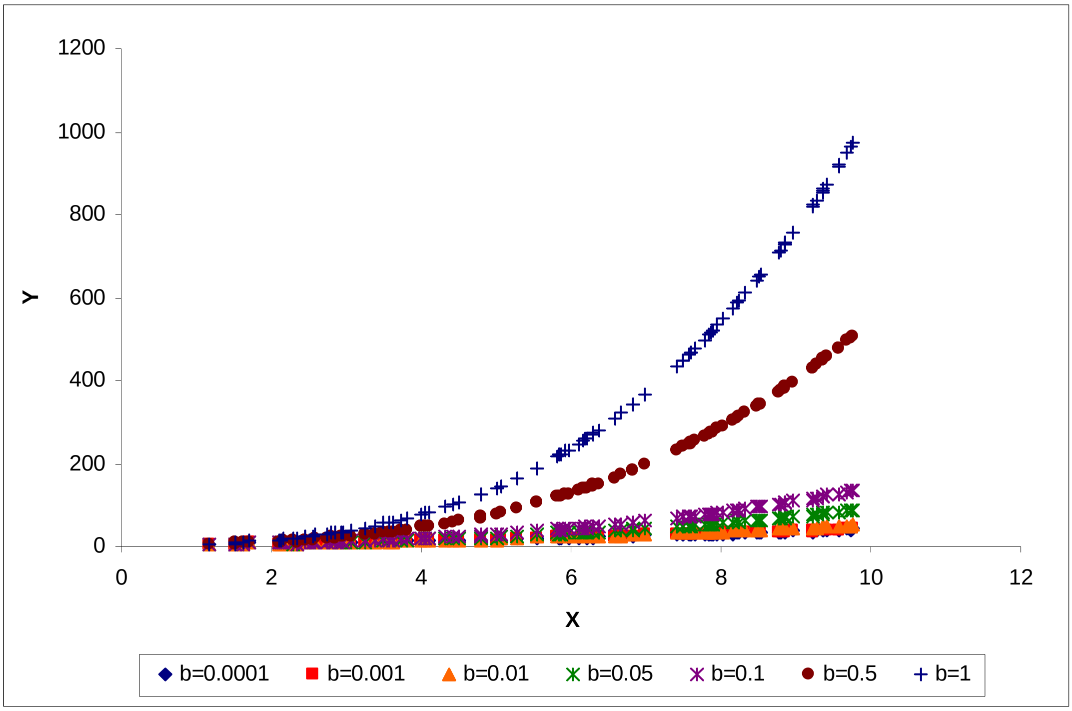
รูปที่ 6 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบเลขชี้กำลัง



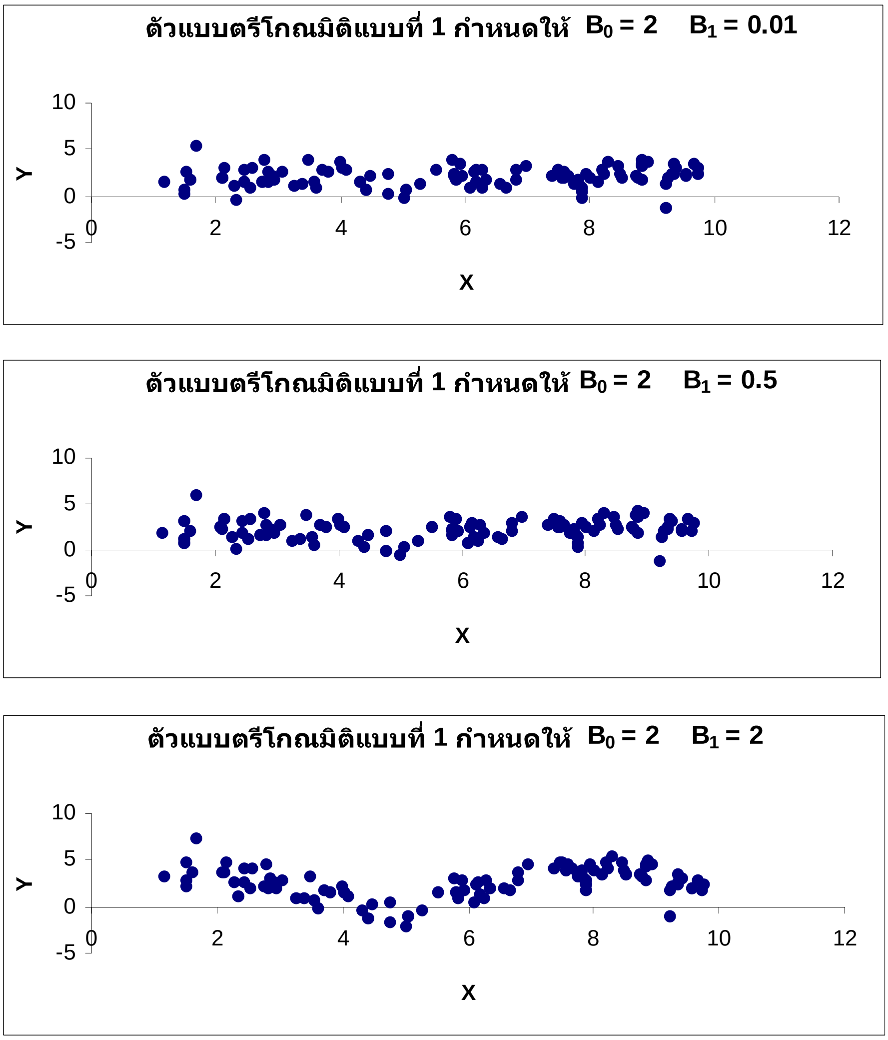
รูปที่ 1 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบเส้นตรง



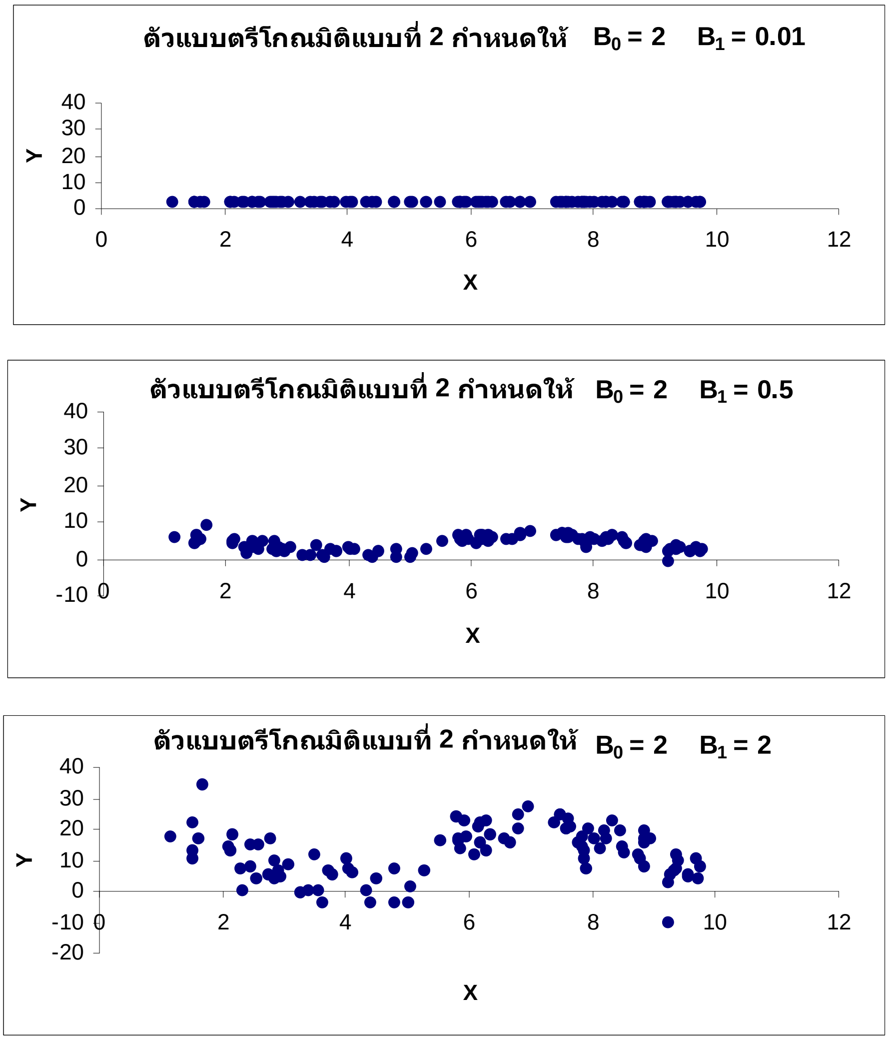
รูปที่ 2 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบพหุนามลำดับที่ 2



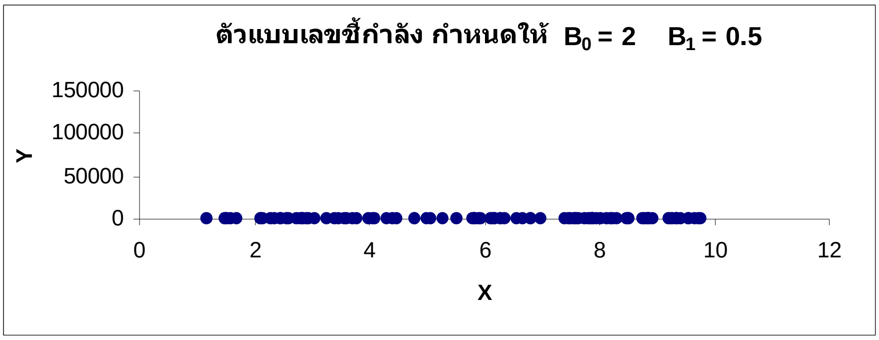
รูปที่ 3 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบพหุนามลำดับที่ 3

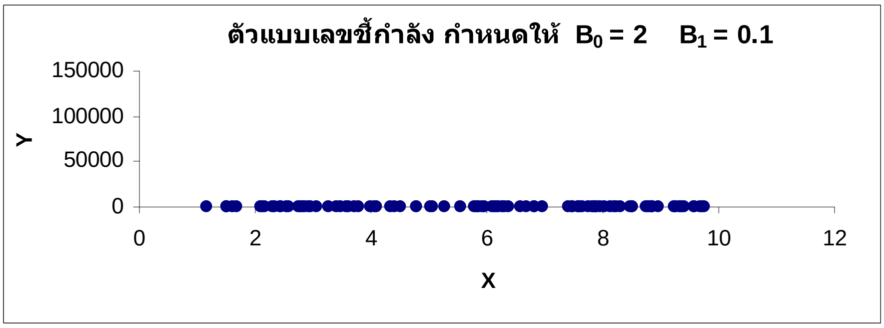


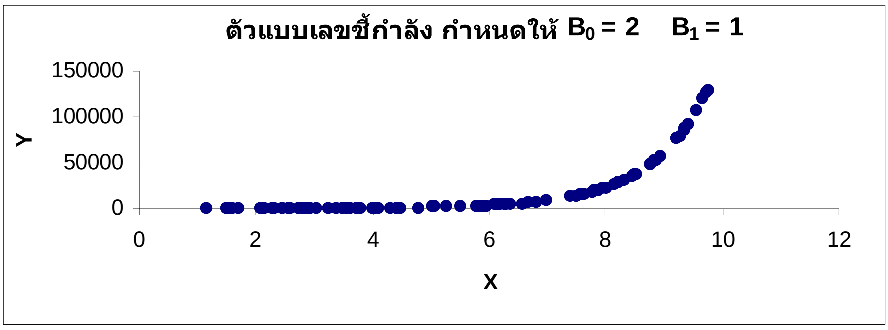
รูปที่ 4 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบตรีโกณมิติแบบที่ 1



รูปที่ 5 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบตรีโกณมิติแบบที่ 2







รูปที่ 6 กราฟแสดงความสัมพันธ์ระหว่าง X กับ Y ที่มีตัวแบบเลขชี้กำลัง