รายการอ้างอิง

ภาษาไทย

เกตุจันทร์ พัชรินทร์ศักดิ์. 2534. การเปรียบเทียบวิธีการนอนพาราเมตริกซ์สำหรับการทดสอบการแจกแจงแบบปกติ. วิทยานิพนธ์ปริญญามหาบัณฑิต คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย.

ยุภาพร รักศิลปกิจ. 2532. การเปรียบเทียบอำนาจการทดสอบของตัวสถิติบางตัวที่ใช้ในการทดสอบหาการแจกแจงแบบยูนิฟอร์ม. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย.

ศิริรัตน์ วงศ์ประกรณ์กุล. 2539. การทดสอบการแจกแจงไวบูลล์และการแจกแจงกอมเฟิรตซ์ ด้วยวิธีทดสอบเทียบความกลมกลืน เมื่อข้อมูลถูกตัดทิ้งอย่างมาก. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย.

สายทอง แจ่มใจ. 2547. การเปรียบเทียบประสิทธิภาพของตัวสถิติทดสอบในการทดสอบภาวะสารูปสนิทดี. วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาสถิติประยุกต์ ภาควิชาสถิติ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร.

ภาษาอังกฤษ

Emad-Eldin A.A.,Mohammed A.S. 1992. On some goodness of fit tests for the Normal , Logistic and Extreme value Distributions. Communications in statistics 21 : 1297-1308.

Jie Mi. 2006. MLE of parameters of location-scale distribution for complete and partially grouped data, Journal of Statistical Planning and Inference 136 : 3565-3582.

Johnson N.L., Kotz S. and Balakrishnan N. 1994. Continuous Univariate Distributions. (Second edition). New York: Wiley,

Lehmann E.L, George Casella. 1998. Theory of Point Estimation. (Second edition). New York: Springer-Verlag,

Ramina Stockute, Andrea Veaux, Paul Johnson. 2006. Logistic Distribution. http://pj.freefaculty.org/stat/Distributions/Logistic-01.pdf

Simon G. Meintanis. 2004. Goodness of fit test for the logistic distribution based on empirical transforms, The Indian Journal of statistics 2 : 306-326.

Stephens M.A. 1979. Test of fit for the logistic Distribution based on the empirical distribution function, Biometrika 66 : 591-595.

ภาคผนวก

การวิจัยครั้งนี้ใช้โปรแกรม R เวอร์ชัน 2.9.2 ในการจำลองข้อมูลและคำนวณค่าต่าง ๆ

โปรแกรมที่ใช้สำหรับการเปรียบเทียบประสิทธิภาพของตัวสถิติที่ใช้สำหรับคัดกรองการแจกแจงแบบโลจิสติกจากการแจกแจงแบบปกติ

```
mu<-0
sigma<-1
scale<-(sqrt(3))/pi
num<-10
N<-5000
alpha<-0.1
ksa<-c();       ksb<- c()
sha<- c();      shb<- c()
ada<- c();      adb<- c()
lia<- c();      lib<- c()
cva<- c();      cvb<- c()
pca<- c();      pcb<- c()

for(n in 1:N)
{
a<-rnorm(num,mu,sigma)                  # generate data from normal distribution
b<-rlogis(num,mu,scale)                 # generate data from logistic distribution

ksa<-c(ksa,ks.test(a,pnorm,0,1)$p.value)    # test type I Komogorov-Smirnov
ksb<-c(ksb,ks.test(b,pnorm,0,1)$p.value)    # test power Komogorov-Smirnov

sha<-c(sha,shapiro.test(a)$p.value)         # test type I Shapiro Wilk
shb<-c(shb,shapiro.test(b)$p.value)         # test power Shapiro Wilk

ada<-c(ada,ad.test(a)$p.value)                  # test type I Anderson Darling
adb<-c(adb,ad.test(b)$p.value)              # test power Anderson Darling
```

```
lia<-c(lia,lillie.test(a)$p.value)          # test type I Lilliefors
lib<-c(lib,lillie.test(b)$p.value)          # test power Lilliefors

cva<-c(cva,cvm.test(a)$p.value)             # test type I Cramer Von Mises
cvb<-c(cvb,cvm.test(b)$p.value)             # test power Cramer Von Mises

pca<-c(pca,pearson.test(a)$p.value)         # test type I Chi-square
pcb<-c(pcb,pearson.test(b)$p.value)         # test power Chi-square
}

ka<-cbind(ksa,sha,ada,lia,cva,pca)
kb<-cbind(ksb,shb,adb,lib,cvb,pcb)
ba1<-ifelse(ka<=0.01,1,0)
ba2<-ifelse(ka<=0.05,1,0)
ba3<-ifelse(ka<=0.1,1,0)

bb1<-ifelse(kb<=0.01,1,0)
bb2<-ifelse(kb<=0.05,1,0)
bb3<-ifelse(kb<=0.1,1,0)

bba<-cbind(ba1,ba2,ba3)
bbb<-cbind(bb1,bb2,bb3)

sa<-apply(bba,2,mean)
sb<-apply(bbb,2,mean)
```

โปรแกรมที่ใช้สำหรับการศึกษาความสัมพันธ์ของการแจกแจงแบบโลจิสติกที่มีค่าเฉลี่ย 0 และความแปรปรวน 1 กับการแจกแจงแบบไคสแควร์ และการแจกแจงแบบที

```
num<-100
N<-5000
mu<-0
scale<-(sqrt(3))/pi
alpha<-0.1
i<-0
j<-0

for(n in 1:N)
{
a<-rlogis(num,mu,scale)                    # generate data from logistic distribution
aa<-sum(a^2)

z<-ks.test(aa,"pchisq",num,0)              # test Chi-square
z1<-z$p.value
i<-ifelse(z1<=alpha,i+1,i)

t1<-mean(a)/(sd(a)/sqrt(num))

d<-ks.test(t1,"pt",num-1)                  # test t
d1<-d$p.value
j<-ifelse(d1<=alpha,j+1,j)
}
i<-i/N
j<-j/N
```

โปรแกรมที่ใช้สำหรับการศึกษาความสัมพันธ์ของการแจกแจงแบบโลจิสติกที่มีค่าเฉลี่ย 0 และความแปรปรวน 1 กับการแจกแจงแบบเอฟ เมื่อข้อมูลเริ่มต้นมี 2 ชุด

```
N<-5000
num1<-25
num2<-25
alpha<-0.1
mu<-0
sigma<-1
scale<-sqrt(3)/pi
j<-0; k<-0; l<-0

for(n in 1:N)
{
a<-rnorm(num1,mu,sigma)                    # generate data from normal distribution
b<-rnorm(num2,mu,sigma)
c<-rlogis(num1,mu,scale)                   # generate data from logistic distribution
d<-rlogis(num2,mu,scale)
f1<-((num1*var(a))/(num1-1))/((num2*var(d))/(num2-1))      # pattern1 N/L
f2<-((num1*var(c))/(num1-1))/((num2*var(b))/(num2-1))      # pattern2 L/N
f3<-((num1*var(c))/(num1-1))/((num2*var(d))/(num2-1))      # pattern3 L/L

j<-ifelse(ks.test(f1,"pf",num1-1,num2-1) $p.value <=alpha,j+1,j)
k<-ifelse(ks.test(f2,"pf",num1-1,num2-1) $p.value <=alpha,k+1,k)
l<-ifelse(ks.test(f3,"pf",num1-1,num2-1) $p.value <=alpha,l+1,l)
}

j<-j/N
k<-k/N
l<-l/N
```

โปรแกรมที่ใช้สำหรับการศึกษาความสัมพันธ์ของการแจกแจงแบบโลจิสติกที่มีค่าเฉลี่ย 0 และความแปรปรวน 1 กับการแจกแจงแบบเอฟ เมื่อข้อมูลเริ่มต้นมาจากข้อมูลชุดเดียว

```
N<-5000
num<-10
alpha<-0.01
nu2<-5
nu1<-5
mu<-0
h<-c()
scale<-(sqrt(3))/pi

for(i in 1: N)
{
        b<-rlogis(num,mu,scale)
        f<-c()

        for(j in 1: num )
        {
                a<-sample(b,nu1)
                d<-setdiff(b,a)
                f1<-var(a)/var(d)
                f<-c(f,f1)
        }

        h<-c(h,ks.test(f,"pf",nu1-1,nu2-1)$p.val)
}
        tran<-ifelse(h<=alpha,1,0)
        ap<-mean(tran)
        ap
```

โปรแกรมที่ใช้สำหรับศึกษาผลกระทบของการประมาณค่าสัมประสิทธิ์การถดถอย เมื่อความคลาดเคลื่อนมีการแจกแจงแบบโลจิสติกโดยการสร้างข้อมูลโดยตรง กรณีการถดถอยอย่างง่าย

```
N<-5000
num<-10
locate<-0
scale<-(sqrt(3))/pi
b0<-1
b1<-2
alpha<-0.05
c1<-c(1,1)
c2<-c(1,10)
c3<-c(10,1)
k<-0;   l<-0
kn<-0; nl<-0
ke<-0; le<-0
belogis<-c()
belogisn<-c()
belogise<-c()

for(n in 1:N)
{
e<-rlogis(num,locate,scale)

        x<-runif(num,-100,100)                  # uniform
        y<-b0+(b1*x)+e
        a<-lsfit(x,y)
                belogis<-rbind(belogis,a$coef)
        h<-lsp(a,1)
        hh<-lsp(a,b1)
```

```
        k<-ifelse(h$coef[[1]][1,4]<=alpha,k+1,k)
        l<-ifelse(hh$coef[[1]][2,4]<=alpha,l+1,l)

xn<-rnorm(num,0,33.34)                    # normal
yn <- b0+(b1*xn)+e
an<-lsfit(xn,yn)
        belogisn<-rbind(belogisn,an$coef)
hn<-lsp(an,b0)
hhn<-lsp(an,b1)
        kn<-ifelse(hn$coef[[1]][1,4]<=alpha,kn+1,kn)
        nl<-ifelse(hhn$coef[[1]][2,4]<=alpha,nl+1,nl)

xe<-rexp(num,0.023)                       # exponential
ye<-b0+(b1*xe)+e
ae<-lsfit(xe,ye)
        belogise<-rbind(belogise,ae$coef)
he<-lsp(ae,b0)
hhe<-lsp(ae,b1)
        ke<-ifelse(he$coef[[1]][1,4]<=alpha,ke+1,ke)
        le<-ifelse(hhe$coef[[1]][2,4]<=alpha,le+1,le)
}

p<-shapiro.test(c1%*%t(belogis))$p.value
q<-shapiro.test(c2%*%t(belogis))$p.value
s<-shapiro.test(c3%*%t(belogis))$p.value
testbelogis<-c(p,q,s)

pn<-shapiro.test(c1%*%t(belogisn))$p.value
qn<-shapiro.test(c2%*%t(belogisn))$p.value
sn<-shapiro.test(c3%*%t(belogisn))$p.value
```

```
        testbelogisn<-c(pn,qn,sn)

        pe<-shapiro.test(c1%*%t(belogise))$p.value
        qe<-shapiro.test(c2%*%t(belogise))$p.value
        se<-shapiro.test(c3%*%t(belogise))$p.value
        testbelogise<-c(pe,qe,se)

typelu<-c(k/N,l/N)
typeln<-c(kn/N,nl/N)
typele<-c(ke/N,le/N)
```

โปรแกรมที่ใช้สำหรับศึกษาผลกระทบของการประมาณค่าสัมประสิทธิ์การถดถอยเมื่อความคลาดเคลื่อนมีการแจกแจงแบบโลจิสติกโดยการสร้างข้อมูลโดยตรง กรณีการถดถอยเชิงพหุ

```
N<-5000
num<-10
mu<-0
scale<-(sqrt(3))/pi
b0<-1
b1<-1
b2<-1
alpha<-0.01
c1<-c(1,1,1)
c2<-c(0.1,1,10)
c3<-c(10,1,0.1)
belogis<-c()
belogisn<-c()
belogise<-c()
k<-0;   kn<-0;  ke<-0
l<-0;   nl<-0;  le<-0
u<-0;   un<-0;  ue<-0

for(n in 1:N)
{
e<-rlogis(num,mu,scale)

        x1<-runif(num,-100,100)                 # uniform
        x2<-runif(num,-100,100)
        x<-cbind(x1,x2)
        y<-0+(b1*x1)+(b2*x2)+e
                a<-lsfit(x,y)
                        belogis<-rbind(belogis,a$coef)
```

```
        h<-lsp(a,1)
        hh<-lsp(a,b1)
                k<-ifelse(h$coef[[1]][1,4]<=alpha,k+1,k)
                l<-ifelse(hh$coef[[1]][2,4]<=alpha,l+1,l)
                u<-ifelse(h$coef[[1]][3,4]<=alpha,u+1,u)

xn1<-rnorm(num,0,33.34)                    #normal
xn2<-rnorm(num,0,33.34)
xn<-cbind(xn1,xn2)
yn<-b0+(b1*xn1)+(b2*xn2)+e
        an<-lsfit(xn,yn)
                belogisn<-rbind(belogisn,an$coef)
        hn<-lsp(an,1)
        hhn<-lsp(an,b1)
                kn<-ifelse(hn$coef[[1]][1,4]<=alpha,kn+1,kn)
                nl<-ifelse(hhn$coef[[1]][2,4]<=alpha,nl+1,nl)
                un<-ifelse(hn$coef[[1]][3,4]<=alpha,un+1,un)

xe1<-rexp(num,0.023)                    #exponential
xe2<-rexp(num,0.023)
xe<-cbind(xe1,xe2)
ye<-b0+(b1*xe1)+(b2*xe2)+e
        ae<-lsfit(xe,ye)
                belogise<-rbind(belogise,ae$coef)
        he<-lsp(ae,1)
        hhe<-lsp(ae,b1)
                ke<-ifelse(he$coef[[1]][1,4]<=alpha,ke+1,ke)
                le<-ifelse(hhe$coef[[1]][2,4]<=alpha,le+1,le)
                ue<-ifelse(he$coef[[1]][3,4]<=alpha,ue+1,ue)
}
```

```
p<-shapiro.test(c1%*%t(belogis))$p.value
q<-shapiro.test(c2%*%t(belogis))$p.value
s<-shapiro.test(c3%*%t(belogis))$p.value
testbelogis<-c(p,q,s)

pn<-shapiro.test(c1%*%t(belogisn))$p.value
qn<-shapiro.test(c2%*%t(belogisn))$p.value
sn<-shapiro.test(c3%*%t(belogisn))$p.value
testbelogisn<-c(pn,qn,sn)

pe<-shapiro.test(c1%*%t(belogise))$p.value
qe<-shapiro.test(c2%*%t(belogise))$p.value
se<-shapiro.test(c3%*%t(belogise))$p.value
testbelogise<-c(pe,qe,se)

typelu<-c(k/N,l/N,u/N)
typeln<-c(kn/N,nl/N,un/N)
typele<-c(ke/N,le/N,ue/N)
```

โปรแกรมที่ใช้สำหรับศึกษาผลกระทบของการประมาณค่าสัมประสิทธิ์การถดถอยเมื่อความคลาดเคลื่อนมีการแจกแจงแบบโลจิสติกโดยการสร้างข้อมูลโดยตรง กรณีการถดถอยเชิงพหุ ตัวแปรอิสระมีความสัมพันธ์กัน

```
N<-5000
num<-10
mu<-0
scale<-(sqrt(3))/pi
b0<-1
b1<-1
b2<-1
alpha<-0.01
c1<-c(1,1,1)
c2<-c(0.1,1,10)
c3<-c(10,1,0.1)

belogis<-c()
belogisn<-c()
belogise<-c()
k<-0;   kn<-0;  ke<-0
l<-0;   nl<-0;  le<-0
u<-0;   un<-0;  ue<-0

for(n in 1:N)
{
e<-rlogis(num,mu,scale)

        z1<-runif(num,-100,100)                    # uniform
        z2<-runif(num,-100,100)
        z3<-runif(num,-100,100)
        x1<-((sqrt(0.9))*z1)+((sqrt(0.1))*z2)
        x2<-((sqrt(0.1))*z2)+((sqrt(0.9))*z3)
```

```
x<-cbind(x1,x2)
y<-0+(b1*x1)+(b2*x2)+e
        a<-lsfit(x,y)
                belogis<-rbind(belogis,a$coef)
        h<-lsp(a,1)
        hh<-lsp(a,b1)
                k<-ifelse(h$coef[[1]][1,4]<=alpha,k+1,k)
                l<-ifelse(hh$coef[[1]][2,4]<=alpha,l+1,l)
                u<-ifelse(h$coef[[1]][3,4]<=alpha,u+1,u)

zn1<-rnorm(num,mu,sigma)                        # normal
zn2<-rnorm(num,mu,sigma)
zn3<-rnorm(num,mu,sigma)
xn1<-((sqrt(0.9))*zn1)+((sqrt(0.1))*zn2)
xn2<-((sqrt(0.1))*zn2)+((sqrt(0.9))*zn3)
xn<-cbind(xn1,xn2)
yn<-b0+(b1*xn1)+(b2*xn2)+e
        an<-lsfit(xn,yn)
                belogisn<-rbind(belogisn,an$coef)
        hn<-lsp(an,1)
        hhn<-lsp(an,b1)
                kn<-ifelse(hn$coef[[1]][1,4]<=alpha,kn+1,kn)
                nl<-ifelse(hhn$coef[[1]][2,4]<=alpha,nl+1,nl)
                un<-ifelse(hn$coef[[1]][3,4]<=alpha,un+1,un)

ze1<-rexp(num,0.023)                        # exponential
ze2<-rexp(num,0.023)
ze3<-rexp(num,0.023)
xe1<-((sqrt(0.9))*ze1)+((sqrt(0.1))*ze2)
xe2<-((sqrt(0.1))*ze2)+((sqrt(0.9))*ze3)
xe<-cbind(xe1,xe2)
```

```
        ye<-b0+(b1*xe1)+(b2*xe2)+e
                ae<-lsfit(xe,ye)
                        belogise<-rbind(belogise,ae$coef)
                he<-lsp(ae,1)
                hhe<-lsp(ae,b1)
                        ke<-ifelse(he$coef[[1]][1,4]<=alpha,ke+1,ke)
                        le<-ifelse(hhe$coef[[1]][2,4]<=alpha,le+1,le)
                        ue<-ifelse(he$coef[[1]][3,4]<=alpha,ue+1,ue)
        }

        p<-shapiro.test(c1%*%t(belogis))$p.value
        q<-shapiro.test(c2%*%t(belogis))$p.value
        s<-shapiro.test(c3%*%t(belogis))$p.value
        testbelogis<-c(p,q,s)

        pn<-shapiro.test(c1%*%t(belogisn))$p.value
        qn<-shapiro.test(c2%*%t(belogisn))$p.value
        sn<-shapiro.test(c3%*%t(belogisn))$p.value
        testbelogisn<-c(pn,qn,sn)

        pe<-shapiro.test(c1%*%t(belogise))$p.value
        qe<-shapiro.test(c2%*%t(belogise))$p.value
        se<-shapiro.test(c3%*%t(belogise))$p.value
        testbelogise<-c(pe,qe,se)

typelu<-c(k/N,l/N,u/N)
typeln<-c(kn/N,nl/N,un/N)
typele<-c(ke/N,le/N,ue/N)
```

โปรแกรมที่ใช้สำหรับศึกษาผลกระทบของการประมาณค่าสัมประสิทธิ์การถดถอยเมื่อความคลาดเคลื่อนมีการแจกแจงแบบโลจิสติกโดยการคัดกรองข้อมูลจากการแจกแจงแบบปกติก่อน กรณีการถดถอยอย่างง่าย

```
R<-450000
N<-5000
num<-10
alpha<-0.01
nu2<-5
nu1<-5
mu<-0
b0<-1
b1<-1
c1<-c(1,1)
c2<-c(1,10)
c3<-c(10,1)
scale<-(sqrt(3))/pi
p<-c(); w<-c()
k<-0;   kn<-0;  ke<-0
l<-0;   nl<-0;  le<-0
belogis<-c()
belogisn<-c()
belogise<-c()

for(i in 1:R)
{

        b<-rlogis(num,mu,scale)
        f<-c()
        for(j in 1: num )
        {
```

```
            a<-sample(b,nu1)
            d<-setdiff(b,a)
            f1<-var(a)/var(d)
            f<-c(f,f1)

        }
        k1<-c(shapiro.test(b)$p.val,ks.test(f,"pf",nu1-1,nu2-1)$p.val)
        if(k1[1]<alpha&&k1[2]<alpha)
            {
                p<-rbind(p,b)
                w<-rbind(w,k1)
            }
}
 for(n in 1:N)
 {
        x<-runif(num,-100,100)                          #uniform
        y<-b0+(b1*x)+p[n,]
            a<-lsfit(x,y)
            belogis<-rbind(belogis,a$coef)
        h<-lsp(a,1)
        hh<-lsp(a,b1)
            k<-ifelse(h$coef[[1]][1,4]<=alpha,k+1,k)
            l<-ifelse(hh$coef[[1]][2,4]<=alpha,l+1,l)

        xn<-rnorm(num,0,33.34)                                  #normal
        yn<-b0+(b1*xn)+p[n,]
            an<-lsfit(xn,yn)
            belogisn<-rbind(belogisn,an$coef)
        hn<-lsp(an,b0)
        hhn<-lsp(an,b1)
```

```
            kn<-ifelse(hn$coef[[1]][1,4]<=alpha,kn+1,kn)
            nl<-ifelse(hhn$coef[[1]][2,4]<=alpha,nl+1,nl)

        xe<-rexp(num,0.023)                                    #exponential
        ye<-b0+(b1*xe)+p[n,]
            ae<-lsfit(xe,ye)
            belogise<-rbind(belogise,ae$coef)
        he<-lsp(ae,b0)
        hhe<-lsp(ae,b1)
            ke<-ifelse(he$coef[[1]][1,4]<=alpha,ke+1,ke)
            le<-ifelse(hhe$coef[[1]][2,4]<=alpha,le+1,le)
}

        pu<-shapiro.test(c1%*%t(belogis))
        q<-shapiro.test(c2%*%t(belogis))
        s<-shapiro.test(c3%*%t(belogis))
        testbelogis<-c(pu$p.value,q$p.value,s$p.value)

        pn<-shapiro.test(c1%*%t(belogisn))
        qn<-shapiro.test(c2%*%t(belogisn))
        sn<-shapiro.test(c3%*%t(belogisn))
        testbelogisn<-c(pn$p.value,qn$p.value,sn$p.value)

        pe<-shapiro.test(c1%*%t(belogise))
        qe<-shapiro.test(c2%*%t(belogise))
        se<-shapiro.test(c3%*%t(belogise))
        testbelogise<-c(pe$p.value,qe$p.value,se$p.value)

pvalu<-c(k/N,l/N)
pvaln<-c(kn/N,nl/N)
pvale<-c(ke/N,le/N)
```

โปรแกรมที่ใช้สำหรับศึกษาผลกระทบของการประมาณค่าสัมประสิทธิ์การถดถอยเมื่อความคลาดเคลื่อนมีการแจกแจงแบบโลจิสติกโดยการคัดกรองข้อมูลจากการแจกแจงแบบปกติก่อน กรณีการถดถอยเชิงพหุ

```
R<-450000
N<-5000
num<-10
alpha<-0.01
nu2<-5
nu1<-5
mu<-0
b0<-1
b1<-1
b2<-1
c1<-c(1,1,1)
c2<-c(0.1,1,10)
c3<-c(10,1,0.1)
scale<-(sqrt(3))/pi
p<-c(); w<-c()
k<-0;   kn<-0;  ke<-0
l<-0;   nl<-0;  le<-0
u<-0;   un<-0;  ue<-0
belogis<-c()
belogisn<-c()
belogise<-c()

for(i in 1:R)
{

      b<-rlogis(num,mu,scale)
      f<-c()
```

```
        for(j in 1: num )
        {
                a<-sample(b,nu1)
                d<-setdiff(b,a)
                f1<-var(a)/var(d)
                f<-c(f,f1)

        }

        k1<-c(shapiro.test(b)$p.val,ks.test(f,"pf",nu1-1,nu2-1)$p.val)
        if(k1[1]<alpha&&k1[2]<alpha)
                {
                        p<-rbind(p,b)
                        w<-rbind(w,k1)
                }
}

for(n in 1:N)
{
        x1<-runif(num,-100,100)                                    #uniform
        x2<-runif(num,-100,100)
        x<-cbind(x1,x2)
        y<-b0+(b1*x1)+(b2*x2)+p[n,]
                a<-lsfit(x,y)
                belogis<-rbind(belogis,a$coef)
        h<-lsp(a,1)
        hh<-lsp(a,b1)
                k<-ifelse(h$coef[[1]][1,4]<=alpha,k+1,k)
                l<-ifelse(hh$coef[[1]][2,4]<=alpha,l+1,l)
                u<-ifelse(h$coef[[1]][3,4]<=alpha,u+1,u)
```

```
        xn1<-rnorm(num,0,33.34)                          # normal
        xn2<-rnorm(num,0,33.34)
        xn<-cbind(xn1,xn2)
        yn<-b0+(b1*xn1)+(b2*xn2)+p[n,]
                an<-lsfit(xn,yn)
                belogisn<-rbind(belogisn,an$coef)
        hn<-lsp(an,1)
        hhn<-lsp(an,b1)
                kn<-ifelse(hn$coef[[1]][1,4]<=alpha,kn+1,kn)
                nl<-ifelse(hhn$coef[[1]][2,4]<=alpha,nl+1,nl)
                un<-ifelse(hn$coef[[1]][3,4]<=alpha,un+1,un)

        xe1<-rexp(num,0.023)                             # exponential
        xe2<-rexp(num,0.023)
        xe<-cbind(xe1,xe2)
        ye<-b0+(b1*xe1)+(b2*xe2)+p[n,]
                ae<-lsfit(xe,ye)
                belogise<-rbind(belogise,ae$coef)
        he<-lsp(ae,1)
        hhe<-lsp(ae,b1)
                ke<-ifelse(he$coef[[1]][1,4]<=alpha,ke+1,ke)
                le<-ifelse(hhe$coef[[1]][2,4]<=alpha,le+1,le)
                ue<-ifelse(he$coef[[1]][3,4]<=alpha,ue+1,ue)
}

        pu<-shapiro.test(c1%*%t(belogis))
        q<-shapiro.test(c2%*%t(belogis))
        s<-shapiro.test(c3%*%t(belogis))
```

```
testbelogis<-c(pu$p.value,q$p.value,s$p.value)

pn<-shapiro.test(c1%*%t(belogisn))
qn<-shapiro.test(c2%*%t(belogisn))
sn<-shapiro.test(c3%*%t(belogisn))
testbelogisn<-c(pn$p.value,qn$p.value,sn$p.value)

pe<-shapiro.test(c1%*%t(belogise))
qe<-shapiro.test(c2%*%t(belogise))
se<-shapiro.test(c3%*%t(belogise))
testbelogise<-c(pe$p.value,qe$p.value,se$p.value)

pvalu<-c(k/N,l/N,u/N)
pvaln<-c(kn/N,nl/N,un/N)
pvale<-c(ke/N,le/N,ue/N)
```

โปรแกรมที่ใช้สำหรับศึกษาผลกระทบของการประมาณค่าสัมประสิทธิ์การถดถอยเมื่อความคลาดเคลื่อนมีการแจกแจงแบบโลจิสติกโดยการคัดกรองข้อมูลจากการแจกแจงแบบปกติก่อน กรณีการถดถอยเชิงพหุ ตัวแปรอิสระมีความสัมพันธ์กัน

```
R<-450000
N<-5000
num<-10
alpha<-0.01
nu2<-5
nu1<-5
mu<-0
b0<-1
b1<-1
b2<-1
c1<-c(1,1,1)
c2<-c(0.1,1,10)
c3<-c(10,1,0.1)
scale<- sqrt(3))/pi
p<-c(); w<-c()
k<-0;   kn<-0;  ke<-0
l<-0;   nl<-0;  le<-0
u<-0;   un<-0;  ue<-0
belogis<-c()
belogisn<-c()
belogise<-c()

for(i in 1:R)
{
        b<-rlogis(num,mu,scale)
        f<-c()
        for(j in 1: num )
```

```
        {
                a<-sample(b,nu1)
                d<-setdiff(b,a)
                f1<-var(a)/var(d)
                f<-c(f,f1)
        }
        k1<-c(shapiro.test(b)$p.val,ks.test(f,"pf",nu1-1,nu2-1)$p.val)
        if(k1[1]<alpha&&k1[2]<alpha)
                {
                        p<-rbind(p,b)
                        w<-rbind(w,k1)
                }
}
for(n in 1:N)
{
        z1<-runif(num,-100,100)                    # uniform
        z2<-runif(num,-100,100)
        z3<-runif(num,-100,100)
        x1<- ((sqrt(0.9))*z1)+((sqrt(0.1))*z2)
        x2<- ((sqrt(0.1))*z2)+((sqrt(0.9))*z3)
        x<-cbind(x1,x2)
        y<-b0+(b1*x1)+(b2*x2)+p[n,]
                a<-lsfit(x,y)
                belogis<-rbind(belogis,a$coef)
        h<-lsp(a,1)
        hh<-lsp(a,b1)
                k<-ifelse(h$coef[[1]][1,4]<=alpha,k+1,k)
                l<-ifelse(hh$coef[[1]][2,4]<=alpha,l+1,l)
                u<-ifelse(h$coef[[1]][3,4]<=alpha,u+1,u)
```

```
zn1<-rnorm(num,0,33.34)                    # normal
zn2<-rnorm(num,0,33.34)
zn3<-rnorm(num,0,33.34)
xn1<- ((sqrt(0.9))*zn1)+((sqrt(0.1))*zn2)
xn2<- ((sqrt(0.1))*zn2)+((sqrt(0.9))*zn3)
xn<-cbind(xn1,xn2)
yn<-b0+(b1*xn1)+(b2*xn2)+p[n,]
        an<-lsfit(xn,yn)
        belogisn<-rbind(belogisn,an$coef)
hn<-lsp(an,1)
hhn<-lsp(an,b1)
        kn<-ifelse(hn$coef[[1]][1,4]<=alpha,kn+1,kn)
        nl<-ifelse(hhn$coef[[1]][2,4]<=alpha,nl+1,nl)
        un<-ifelse(hn$coef[[1]][3,4]<=alpha,un+1,un)

ze1<-rexp(num,0.023)                    # exponential
ze2<-rexp(num,0.023)
ze3<-rexp(num,0.023)
xe1<- ((sqrt(0.9))*ze1)+((sqrt(0.1))*ze2)
xe2<- ((sqrt(0.1))*ze2)+((sqrt(0.9))*ze3)
xe<-cbind(xe1,xe2)
ye<-b0+(b1*xe1)+(b2*xe2)+p[n,]
        ae<-lsfit(xe,ye)
        belogise<-rbind(belogise,ae$coef)
he<-lsp(ae,1)
hhe<-lsp(ae,b1)
        ke<-ifelse(he$coef[[1]][1,4]<=alpha,ke+1,ke)
        le<-ifelse(hhe$coef[[1]][2,4]<=alpha,le+1,le)
        ue<-ifelse(he$coef[[1]][3,4]<=alpha,ue+1,ue)
}
```

```
        pu<-shapiro.test(c1%*%t(belogis))
        q<-shapiro.test(c2%*%t(belogis))
        s<-shapiro.test(c3%*%t(belogis))
        testbelogis<-c(pu$p.value,q$p.value,s$p.value)

        pn<-shapiro.test(c1%*%t(belogisn))
        qn<-shapiro.test(c2%*%t(belogisn))
        sn<-shapiro.test(c3%*%t(belogisn))
        testbelogisn<-c(pn$p.value,qn$p.value,sn$p.value)

        pe<-shapiro.test(c1%*%t(belogise))
        qe<-shapiro.test(c2%*%t(belogise))
        se<-shapiro.test(c3%*%t(belogise))
        testbelogise<-c(pe$p.value,qe$p.value,se$p.value)

pvalu<-c(k/N,l/N,u/N)
pvaln<-c(kn/N,nl/N,un/N)
pvale<-c(ke/N,le/N,ue/N)
```

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวสุภาวดี วิชิตชาญ เกิดเมื่อวันที่ 30 สิงหาคม พ.ศ. 2529 สำเร็จการศึกษาระดับปริญญาวิทยาศาสตรบัณฑิต (วท.บ.) สถิติ จากภาควิชาสถิติ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร ในปีการศึกษา 2550 และเข้าศึกษาต่อในหลักสูตรสถิติศาสตรมหาบัณฑิต (สต.ม.) สาขาสถิติ ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2551