รายการอ้างอิง

ภาษาไทย

ธีระพร วีระถาวร. ตัวแบบเชิงเส้น:ทฤษฎีและการประยุกต์. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : บริษัทพิมพ์ดีจำกัด , 2541.

มานพ วราภักดิ์. การจำลองเบื้องต้น. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : ศูนย์ผลิตตำราเรียน สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ , 2547.

วิลาสินี จันทราวุฒิ. การเปรียบเทียบวิธีการประมาณค่าพารามิเตอร์สำหรับแผนแบบสุ่มตลอดในบล็อกสมบูรณ์อิทธิพลคงที่ กรณีข้อมูลระยะยาว. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ,2545.

ศศิธร เจษฎาฐิติกุล. การเปรียบเทียบการทดสอบความเป็นอิสระระหว่างตัวแปร 2 ตัวแปรเมื่อตัวแปรอิสระมีการแจกแจงพหุนามและอยู่ในตารางการณ์จร 2 ทาง. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ,2545.

อรไท สงวนสินธ์. การเปรียบเทียบการทดสอบเอฟและการทดสอบมอนติคาร์โลด้วยอัตราส่วนภาวะน่าจะเป็นสำหรับแผนการทดลองแบบสุ่มตลอดที่ปัจจัยทดลองคงที่. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ,2545.

ภาษาอังกฤษ

Manly ,Bryan F.J. Randomization ,Bootstrap and Monte Carlo method in Biology. 2nd Edition. Boca Raton : Chapman & Hall. ,1997.

Seber,G.A.F. Linear Regression Analysis. New York: John Wiley & Sons ,1977.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บรรณานุกรม

ภาษาไทย

คนึงนิจ เสรีวงษ์. การวิเคราะห์ความถดถอย. กรุงเทพมหานคร : ภาควิชาคณิตศาสตร์และสถิติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์, 2543. (อัดสำเนา)

ธีระพร วีระถาวร. การอนุมานเชิงสถิติขั้นกลาง:โครงสร้างและความหมาย. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย , 2536.

ประชุม สุวัตถี. ทฤษฎีการอนุมานเชิงสถิติ. พิมพ์ครั้งที่ 2 ปรับปรุงแก้ไข. กรุงเทพมหานคร : โครงการส่งเสริมเอกสารวิชาการ สถาบันบัณฑิตพัฒนบริหารศาสตร์ , 2545.

สมจิต วัฒนาชยากูล. สถิติวิเคราะห์เบื้องต้น. พิมพ์ครั้งที่ 3. กรุงเทพมหานคร : สำนักพิมพ์ประกายพรึก , 2532.

ภาษาอังกฤษ

Hogg,Robert V. and Craig,Allen T. Introduction to Mathematical Statistics. 5th edition. New Jersey : Prentice – Hall ,1995.

Jean-Marie Dufour y and Lynda Khalaf z. Monte Carlo Test Methods in Econometrics. Available from: http://www.fas.umontreal.ca/SECCO/dufour/Dufour_Khalaf_1998_MCTcompanion_W.pdf

Ronald R. Hocking. Methods and Applications of Linear Model:Regression and the Analysis of variance. 1st edition New York :John Wiley & Sons ,1996.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางแสดงฟังก์ชันการทำงานของโปรแกรม S-PLUS 2000 ที่ใช้ในการวิจัย

ฟังก์ชัน	หน้าที่การทำงาน
rmvnorm(n,mean=rep(0,d),cov=diag(d),sd,rho)	ทำการสร้างเมตริกซ์ตัวเลขสุ่มให้มีการแจกแจงแบบปกติหลายตัวแปรโดยกำหนดขนาดตัวอย่าง(จำนวนแถว) กำหนดค่าเฉลี่ยของแต่ละตัวแปรและกำหนดจำนวนตัวแปร(จำนวนสดมภ์)กำหนดเมทริกซ์ความแปรปรวนร่วมเป็นเมทริกซ์เอกลักษณ์คือ sd และกำหนดความสัมพันธ์ระหว่างตัวแปรให้เป็น rho
cbind(1,x)	ทำการเพิ่มสดมภ์อีก 1 สดมภ์ที่มีค่าเป็น 1 ของเมทริกซ์ x
select.cols(indepen,1:(inde+1))	ทำการเลือกสดมภ์จากเมทริกซ์ indepen โดยเลือกจากสดมภ์ที่ 1 จนถึงสดมภ์ที่ inde+1
Matrix(a,ncol=3, byrow =T)	ทำการสร้างเมทริกซ์จากตัวแปรเวกเตอร์ a โดยกำหนดให้มี 3 สดมภ์
ifelse	การเลือกชุดข้อมูลมาตามเงื่อนไขที่ตั้งไว้
dim	กำหนดขนาดของเวกเตอร์ที่ใช้ในการเก็บข้อมูล
array(c(),dim)	ทำการเก็บข้อมูลในรูปเวกเตอร์ โดยจะใช้คู่กับ dim
rnorm	ทำการสร้างตัวเลขสุ่มให้มีการแจกแจงแบบปกติ
as.matrix	กำหนดตัวแปรที่เป็นเวกเตอร์ให้เป็นเมทริกซ์
mean	คำนวณหาค่าเฉลี่ยของข้อมูล
stdev	คำนวณหาค่าเบี่ยงเบนมาตรฐานของข้อมูล
dig	การกำหนดทศนิยมที่ต้องการ
round(y,dig)	ทำการปัดเศษของข้อมูลที่ y โดยจะใช้คู่กับ dig
as.vector	กำหนดตัวแปรที่เป็นเมทริกซ์ให้เป็นเวกเตอร์
sum	หาผลรวมของข้อมูล
ifelse	การเลือกชุดข้อมูลมาตามเงื่อนไขที่ตั้งไว้

ตารางแสดงความหมายของสัญลักษณ์ต่าง ๆ ของโปรแกรม S-PLUS 2000

สัญลักษณ์	ความหมาย
Inde	จำนวนตัวแปรอิสระที่ใช้ในการทดลอง
sd	ค่าเบี่ยงเบนมาตรฐาน
n	จำนวนขนาดตัวอย่าง
k	ความแตกต่างระหว่างสัมประสิทธิ์ความถดถอย
loops	จำนวนรอบของการทำซ้ำในแต่ละสถานการณ์
f.stat	ค่าสถิติทดสอบเอฟที่คำนวณได้
RL.pvalue, MCL.pvalue	ค่า p-value ของตัวสถิติทดสอบอัตราส่วนความควรจะเป็นแบบปกติ และตัวสถิติของทดสอบอัตราส่วนความควรจะเป็นแบบมอนติคาร์โล
lamda.reg, lamda.monte	ค่าสถิติทดสอบอัตราส่วนความควรจะเป็นแบบปกติและที่ได้จากการทดสอบอัตราส่วนความควรจะเป็นแบบมอนติคาร์โล
trials	จำนวนรอบที่ทำซ้ำในการทดสอบอัตราส่วนความควรจะเป็นแบบมอนติคาร์โล
prob.li0.01, prob.li0.05, prob.li0.1	ค่าสัดส่วนของการปฏิเสธสมมติฐานว่าง/ค่าอำนาจการทดสอบจากการทดสอบอัตราส่วนความควรจะเป็นแบบปกติที่ระดับนัยสำคัญ 0.01 0.05 และ 0.1
prob.lim0.01, prob.lim0.05, prob.lim0.1	ค่าสัดส่วนของการปฏิเสธสมมติฐานว่าง/ค่าอำนาจการทดสอบจากการทดสอบอัตราส่วนความควรจะเป็นแบบมอนติคาร์โลที่ระดับนัยสำคัญ 0.01 0.05 และ 0.1

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

โปรแกรมการทดสอบสมมติฐานเกี่ยวกับสัมประสิทธิ์ความถดถอยของการทดสอบอัตราส่วนความควรจะเป็นแบบปกติและการทดสอบอัตราส่วนความควรจะเป็นแบบมอนติคาร์โล

```
/*****************************************************************************************************/
/*                    การกำหนดค่าสถานการณ์ต่าง ๆภายใต้สมมติฐานว่าง                                  */
/*****************************************************************************************************/
inde <- 2
sd <- 1
n1<-10
trials <- 100
loops <- 1500

#ฟังก์ชันการสร้างตัวแปรอิสระ
      x_rmvnorm(100,mean=c(rep(0,inde)),cov=diag(1,inde),rho=0)
      indepen_cbind(1,x)
      xmat_select.cols(indepen, 1:(inde+1))
      txmat_t(xmat)

#สร้างค่าคงที่
      maA <- function(inde)
      {
            if (inde==2)      {a <- c(1,0,0,0,1,0,0,0,1)
                  aa <- matrix(a,ncol=3,byrow=T)
            }else if(inde==3)  {a <- c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1)
                  aa <- matrix(a,ncol=4,byrow=T)
            }else if(inde==4)   {a <- c(1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1)
                  aa <- matrix(a,ncol=5,byrow=T)
            }else if(inde==5)   {a <-
      c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1)
                  aa <- matrix(a,ncol=6,byrow=T)
            }else stop("missing value")
```

```
        }
        matA <- maA(inde)
        maC <- function(inde)
        {
                if (inde==2)        {c <- c(10,10,10)
                        cc <- matrix(c,ncol=1,byrow=T)
                }else if(inde==3)  {c <- c(10,10,10,10)
                        cc <- matrix(c,ncol=1,byrow=T)
                }else if(inde==4)   {c <- c(10,10,10,10,10)
                        cc <- matrix(c,ncol=1,byrow=T)
                }else if(inde==5)   {c <- c(10,10,10,10,10,10)
                        cc <- matrix(c,ncol=1,byrow=T)
                }else stop("missing value")
        }
        matC <- maC(inde)
#n=10
{
        txN10_select.cols(txmat, 1:n1)
        xN10_t(txN10)
#keep value of Regular likelihood ratio test
        RL.pvalueN10 <- array(,dim=c(1,loops))
#keep value of Monte Carlo likelihood ratio test
        MCL.pvalueN10 <- array(,dim=c(1,loops))

        for(l in 1:loops)
        {
                erN10 <- rnorm(n1,0,sd)
                errorN10 <- as.matrix(erN10)
                m1N10 <- array(,dim=c(1))
                sd1N10 <- array(,dim=c(1))
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
#ฟังก์ชันการกำหนดตัวแปรตาม
        yN10 <- xN10%*%matC+errorN10
        m1 <- mean(yN10)
        sd1 <- stdev(yN10)
#คำนวณค่า Beta
        txN10 <- t(xN10)%*%xN10
        itxN10 <- solve(txN10)
        betaN10 <- itxN10%*%(t(xN10)%*%yN10)
#การทดสอบสมมติฐาน
        aiaN10 <- (matA)%*%itxN10%*%t(matA)
        iaiaN10 <- solve(aiaN10)
        qN10 <- t(matA%*%betaN10-matC)%*%(iaiaN10)%*%(matA%*%betaN10-
matC)
        ssN10 <- yN10 - (xN10%*%betaN10)
        sseN10 <- t(ssN10)%*%ssN10
        lamdaN10 <- (1/(1+(qN10/sseN10)))^(n1/2)
        lamdaN10 <- round(lamdaN10,dig=8)
        lamda.regN10 <- as.vector(lamdaN10)
        dfr <- inde
        f.statN10 <- (qN10/dfr)/(sseN10/(n1-inde-1))
        f.statN10 <- round(f.statN10,dig=8)
        RL.pvalueN10[,l] <- round(1-pf(f.statN10,dfr,n1-inde-1),dig=8)
#การทดสอบสมมติฐานโดยการทดสอบอัตราส่วนความควรจะเป็นแบบปกติ
        Lamda.monteN10 <- array(,dim=c(1,trials))
            for(i in 1:trials)
            {
                emonN10 <- rnorm(n1,0,sd)
                ymonN10 <- xN10%*%matC+emonN10
                ymonN10
            #คำนวณค่า Beta
```

```
                    beta.monN10 <- itxN10%*%(t(xN10)%*%ymonN10)
                #การทดสอบสมมติฐาน
                    qmonN10 <- t(matA%*%beta.monN10-
    matC)%*%(iaiaN10)%*%(matA%*%beta.monN10-matC)
                    ssmonN10 <- ymonN10 - (xN10%*%beta.monN10)
                    sse.monN10 <- t(ssmonN10)%*%ssmonN10
                    pmonN10 <- (1+(qmonN10/sse.monN10))
                    Lamda.monteN10[,i] <- round(((1/pmonN10)^(n1/2)),dig=8)
                    Lamda.monteN10
            }
            # Compute p-value of Monte carlo likelihood ratio test At N=10
                countN10 <- ifelse(Lamda.monteN10 <= lamda.regN10,1,0)
                sumli.ratioN10 <- sum(countN10)
                MCL.pvalueN10[,I] <- round(sumli.ratioN10/trials,dig=5)
    }
}
#N10
    rlN10 <- t(RL.pvalueN10)
    mclN10 <- t(MCL.pvalueN10)
    histomclN10_hist(mclN10)
    historlN10_hist(rlN10)
    #Compute proportion p-value of regular likelihood ratio test at 0.01
        N10count.li0.01 <- ifelse(RL.pvalueN10<=0.01,1,0)
        N10sum.pvalue0.01 <- sum(N10count.li0.01)
        N10prob.li0.01 <- round(N10sum.pvalue0.01/loops,dig=8)
        N10prob.li0.01
    #Compute proportion p-value of regular likelihood ratio test at 0.05
        N10count.li0.05 <- ifelse(RL.pvalueN10<=0.05,1,0)
        N10sum.pvalue0.05 <- sum(N10count.li0.05)
        N10prob.li0.05 <- round(N10sum.pvalue0.05/loops,dig=8)
```

```
N10prob.li0.05
#Compute proportion p-value of regular likelihood ratio test at 0.1
N10count.li0.1 <- ifelse(RL.pvalueN10<=0.1,1,0)
N10sum.pvalue0.1 <- sum(N10count.li0.1)
N10prob.li0.1 <- round(N10sum.pvalue0.1/loops,dig=8)
N10prob.li0.1
#Compute proportion p-value of monte carlo likelihood ratio test at 0.01
N10count.lim0.01 <- ifelse(MCL.pvalueN10<=0.01,1,0)
N10sum.mpvalue0.01 <- sum(N10count.lim0.01)
N10prob.lim0.01 <- round(N10sum.mpvalue0.01/loops,dig=8)
N10prob.lim0.01
#Compute proportion p-value of monte carlo likelihood ratio test at 0.05
N10count.lim0.05 <- ifelse(MCL.pvalueN10<=0.05,1,0)
N10sum.mpvalue0.05 <- sum(N10count.lim0.05)
N10prob.lim0.05 <- round(N10sum.mpvalue0.05/loops,dig=8)
N10prob.lim0.05
#Compute proportion p-value of monte carlo likelihood ratio test at 0.1
N10count.lim0.1 <- ifelse(MCL.pvalueN10<=0.1,1,0)
N10sum.mpvalue0.1 <- sum(N10count.lim0.1)
N10prob.lim0.1 <- round(N10sum.mpvalue0.1/loops,dig=8)
N10prob.lim0.1
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
##########################################################################
                    END    PROGRAM                                      #
##########################################################################
```

```
/*****************************************************************************************************/
/*                    การกำหนดค่าสถานการณ์ต่าง ๆภายใต้สมมติฐานแย้ง                                  */
/*****************************************************************************************************/
inde <- 2
sd <- 1
n1 <- 10
trials <- 100
loops <- 1500
k <- 0.05*sd
#ฟังก์ชันการสร้างตัวแปรอิสระ
      x_rmvnorm(100,mean=c(rep(0,inde)),cov=diag(1,inde),rho=0)
      indepen_cbind(1,x)
      xmat_select.cols(indepen, 1:(inde+1))
      tx_t(xmat)
#สร้างสัมประสิทธิ์ความถดถอย
maB <- function(inde)
{
      if (inde==2)        {b <- c(10,10,10+k)
            bb <- matrix(b,ncol=1,byrow=T)
      }else if(inde==3)   {b <- c(10,10,10,10+k)
            bb <- matrix(b,ncol=1,byrow=T)
      }else if(inde==4)   {b <- c(10,10,10,10,10+k)
            bb <- matrix(b,ncol=1,byrow=T)
      }else if(inde==5)   {b <- c(10,10,10,10,10,10+k)
            bb <- matrix(b,ncol=1,byrow=T)
      }else stop("missing value")
}
matB <- maB(inde)
#สร้างค่าคงที่
maA <- function(inde)
```

```
 {
        if (inde==2)        {a <- c(1,0,0,0,1,0,0,0,1)
              aa <- matrix(a,ncol=3,byrow=T)
        }else if(inde==3)  {a <- c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1)
              aa <- matrix(a,ncol=4,byrow=T)
        }else if(inde==4)   {a <- c(1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1)
              aa <- matrix(a,ncol=5,byrow=T)
        }else if(inde==5)   {a <-
        c(1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1)
              aa <- matrix(a,ncol=6,byrow=T)
        }else stop("missing value")
}
matA <- maA(inde)
maC <- function(inde)
{
        if (inde==2)        {c <- c(10,10,10)
              cc <- matrix(c,ncol=1,byrow=T)
        }else if(inde==3)  {c <- c(10,10,10,10)
              cc <- matrix(c,ncol=1,byrow=T)
        }else if(inde==4)   {c <- c(10,10,10,10,10)
              cc <- matrix(c,ncol=1,byrow=T)
        }else if(inde==5)   {c <- c(10,10,10,10,10,10)
              cc <- matrix(c,ncol=1,byrow=T)
        }else stop("missing value")
}
matC <- maC(inde)
#n=10
{
      {
        txN10_select.cols(tx, 1:n1)
```

```
        xN10_t(txN10)
#keep value of Regular likelihood ratio test
        RL.powerN10 <- array(,dim=c(1,loops))
#keep value of Monte Carlo likelihood ratio test
        MCL.powerN10 <- array(,dim=c(1,loops))
#คำนวณค่า Beta
        txN10 <- t(xN10)%*%xN10
        itxN10 <- solve(txN10)
        betaN10 <- matB
#การทดสอบสมมติฐาน
        aia <- (matA)%*%itxN10%*%t(matA)
        iaia <- solve(aia)
        qN10 <- t(matA%*%betaN10-matC)%*%(iaia)%*%(matA%*%betaN10-matC)
for(I in 1:loops)
{
        erN10 <- rnorm(n1,0,sd)
        error <- as.matrix(erN10)
        m1 <- array(,dim=c(1))
        sd1 <- array(,dim=c(1))
#ฟังก์ชันการกำหนดตัวแปรตาม
        yN10 <- xN10%*%matB+error
        m1 <- mean(yN10)
        sd1 <- stdev(yN10)
        ss <- yN10 - (xN10%*%betaN10)
        sse <- t(ss)%*%ss
        lamda <- (1/(1+(qN10/sse)))^(n1/2)
        lamda <- round(lamda,dig=8)
        lamda.reg <- as.vector(lamda)
        dfr <- inde
        f.stat <- (qN10/dfr)/(sse/(n1-inde-1))
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
            f.stat <- round(f.stat,dig=8)
            RL.powerN10[,l] <- round(1-pf(f.stat,dfr,n1-inde-1),dig=8)
            Lamda.monte <- array(,dim=c(1,trials))
        #คำนวณค่า Beta
            beta.monN10 <- matB
        #การทดสอบสมมติฐาน
            qmonN10 <- t(matA%*%beta.monN10-
        matC)%*%(iaia)%*%(matA%*%beta.monN10-matC)
            for(i in 1:trials)
            {
                emon <- rnorm(n1,0,sd)
                ymonN10 <- xN10%*%betaN10+emon
                ymonN10
                ssmon <- ymonN10 - (xN10%*%beta.monN10)
                sse.mon <- t(ssmon)%*%ssmon
                pmon <- (1+(qmonN10/sse.mon))
                Lamda.monte[,i] <- round(((1/pmon)^(n1/2)),dig=8)
                Lamda.monte
            }
        # Compute power of Monte carlo likelihood ratio test At N=10
            count <- ifelse(Lamda.monte <= lamda.reg,1,0)
            sumli.ratio <- sum(count)
            MCL.powerN10[,l] <- round(sumli.ratio/trials,dig=5)
        }
        rlN10 <- t(RL.powerN10)
        mclN10 <- t(MCL.powerN10)
        histomclN10_hist(mclN10)
        historlN10_hist(rlN10)
}
#Compute proportion power of regular likelihood ratio test at 0.01
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
        N10count.li0.01 <- ifelse(RL.powerN10<=0.01,1,0)
        N10sum.power0.01 <- sum(N10count.li0.01)
        N10prob.li0.01 <- round(N10sum.power0.01/loops,dig=8)
#Compute proportion power of regular likelihood ratio test at 0.05
        N10count.li0.05 <- ifelse(RL.powerN10<=0.05,1,0)
        N10sum.power0.05 <- sum(N10count.li0.05)
        N10prob.li0.05 <- round(N10sum.power0.05/loops,dig=8)
#Compute proportion power of regular likelihood ratio test at 0.1
        N10count.li0.1 <- ifelse(RL.powerN10<=0.1,1,0)
        N10sum.power0.1 <- sum(N10count.li0.1)
        N10prob.li0.1 <- round(N10sum.power0.1/loops,dig=8)
#Compute proportion power of monte carlo likelihood ratio test at 0.01
        N10count.lim0.01 <- ifelse(MCL.powerN10<=0.01,1,0)
        N10sum.mpower0.01 <- sum(N10count.lim0.01)
        N10prob.lim0.01 <- round(N10sum.mpower0.01/loops,dig=8)
#Compute proportion power of monte carlo likelihood ratio test at 0.05
        N10count.lim0.05 <- ifelse(MCL.powerN10<=0.05,1,0)
        N10sum.mpower0.05 <- sum(N10count.lim0.05)
        N10prob.lim0.05 <- round(N10sum.mpower0.05/loops,dig=8)
#Compute proportion power of monte carlo likelihood ratio test at 0.1
        N10count.lim0.1 <- ifelse(MCL.powerN10<=0.1,1,0)
        N10sum.mpower0.1 <- sum(N10count.lim0.1)
        N10prob.lim0.1 <- round(N10sum.mpower0.1/loops,dig=8)
}

##############################################################################
#                          END    PROGRAM                                    #
##############################################################################
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายวโรภาส ประดิษฐกำจรชัย เกิดวันที่ 30 พฤศจิกายน พ.ศ.2520 จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิทยาศาสตรบัณฑิต (วท.บ.) วิชาเอกสถิติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2543 และเข้าศึกษาต่อในหลักสูตรสถิติศาสตรมหาบัณฑิต (สต.ม.) ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ.2544

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย