รายการอ้างอิง

<u>ภาษาไทย</u>

ธิดาเดียว มยุรีสวรรค์."การเปรียบเทียบวิธีการประมาณช่วงความเชื่อมั่นสำหรับค่าสัดส่วนประชากร" วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2535.

ธีระพร วีระถาวร.<u>การอนุมานสถิติขั้นกลาง: โครงสร้างและความหมาย</u>. กรุงเทพมหานคร: ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย,2536.

ธีระพร วีระถาวร. <u>ตัวแบบเชิงเส้น: ทฤษฎีและการประยุกต์</u>. กรุงเทพมหานคร: ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย,2541.

มาลี ตระการศีรินนท์. "การเปรียบเทียบการประมาณค่าพารามิเตอร์ของตัวแบบสมการความถดถอยเชิงเส้นด้วยวิธีกำลังสองต่ำสุดและวิธีบูตสแตรป" วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2532.

รวมพร ทองรัศมี. "การเปรียบเทียบการทดสอบเทียบความกลมกลืนสำหรับการแจกแจงแบบเลขชี้กำลังที่มีสองพารามิเตอร์" วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2541.

วิชิต หล่อจีระชุณห์กุล และคณะ.<u>เทคนิคการพยากรณ์</u>.พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : โครงการส่งเสริมเอกสารวิชาการ สถาบันบัณฑิตพัฒนบริหารศาสตร์, 2539.

ศิริรัตน์ วงศ์ประกรณ์กุล. "การทดสอบการแจกแจงไวบูลล์และการแจกแจงคอมเพิรตซ์ด้วยวิธีทดสอบเทียบความกลมกลืนเมื่อข้อมูลถูกตัดทิ้งอย่างมาก" วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาการประกันภัย บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2538.

<u>ภาษาอังกฤษ</u>

Abraham, B., and Ledolter, J. <u>Statistical Methods for Forecasting.</u> New York : John Wiley & sons, 1983.

Conover, W.J. <u>Practical Nonparametric Statistics.</u> 2nd ed., New York : John Wiley & sons , 1980.

Daniel, W. W. <u>Biostatistics : a foundation for analysis in the health sciences</u>,4th ed. New York : Wiley, 1995.

Efron, B., and Tibshirani, R. <u>An Introduction to the Bootstrap.</u> London : Chapman & Hall, 1993.

Efron, B. "Bootstrap Methods: Another Look at the Jackknife." The Annals of Statistics, 7 (1979) : 1-26.

Hardle, W., and Mammen, E. "Comparing Nonparaametric Versus Parametric Regression Fits." The Annals of Statistics, 21 (1993) :1926-1947.

Law, a. W. and W. D. Kelton. Simulation Modeling and Analysis. 2^{nd} ed., Singgapore : McGraw-Hill, 1991.

Seber, G. A. Ff. Liner Regression Analysis. New York : John Wiley & Sons, 1977.

Stute, W. "Nonparametric Model Checks for Regression." The Annals of Statistics, 25 (1997) : 613-641.

Stute, W., W. Gonzalez Manteiga and M. Presedo Quindimil. "Bootstrap Approximations in Model Checks for Regression." Journal of the American Statistical Association, 93 (1998) : 141-149.

Su, J. Q., and Wei, L. J. "A Lack-of-Fit Test for the Mean Function in a Generalized Linear Model." Journal of the American Statistical Association, 86 (1991) : 420-426.

Wu, C. F. J. "Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis." The Annals of Statistics, 14 (1986) :1261-1295.

ภาคผนวก

ภาคผนวก ก

การผลิตข้อมูลที่ใช้ในการวิจัย

1. การสุ่มตัวอย่างแบบใส่คืน (Sampling with replacement)

เป็นการสุ่มตัวอย่างที่ยอมให้มีหน่วยตัวอย่างซ้ำกันได้ นั่นคือแต่ละหน่วยตัวอย่างมีโอกาส (probability)ในการถูกสุ่มเท่ากัน คือ 1/N เมื่อ N คือขนาดของประชากร ในการวิจัยครั้งนี้ได้ใช้เครื่องคอมพิวเตอร์ช่วยในการสุ่มตัวอย่างแบบใส่คืน โดยใช้ตัวเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง [0,1] เป็นตัวเปรียบเทียบกับค่าความน่าจะเป็นสะสม(Cumulative Probability) เพื่อกำหนดการสุ่มหน่วยตัวอย่างให้มีขนาดเท่ากับขนาดตัวอย่างที่มีอยู่ ซึ่งขั้นตอนการสุ่มตัวอย่างแบบใส่คืนสามารถสรุปได้พอสังเขปดังนี้

1. คำนวณค่าความน่าจะเป็นของแต่ละหน่วยตัวอย่าง = 1/N
2. หาความน่าจะเป็นสะสมแล้วจัดช่วง
3. สร้างตัวเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง [0,1]
4. นำตัวเลขสุ่มในขั้นตอนที่ 3 มาเปรียบเทียบกับค่าความน่าจะเป็นสะสม ถ้าตกอยู่ช่วงใดหน่วยนั้นๆ จะถูกเลือกมาเป็นตัวอย่าง
5. กระทำซ้ำขั้นตอนที่ 3 และ 4 จำนวน n ครั้ง เมื่อ n คือขนาดตัวอย่างที่ต้องการ

จากขั้นตอนข้างต้น โปรแกรมย่อยที่ใช้ในการสุ่มตัวอย่างแบบใส่คืนแสดงได้ดังนี้

```
******************************************************************************
*               Subroutine for Sampling with Replacement                    *
******************************************************************************
      SUBROUTINE SWR(N,EBS)
      DIMENSION P(100),E(100),EBS(100),X(100,10)
      COMMON/SAMPLE/X,E/PROB/P/SEED/IX,KN
      DO 300 J=1,N
      CRN=RAND(IX)
      DO 305 I=1,N
      II=I-1
      IF (II.EQ.0) THEN
      X1=0.0
      ELSE
      X1=P(II)
```

```
      END IF
      X2=P(I)
      IF((CRN.GT.X1).AND.(CRN.LE.X2)) THEN
      EBS(J)=E(I)
      GOTO 300
      END IF
305   CONTINUE
300   CONTINUE
      RETURN
      END
```

เมื่อ N เป็นขนาดตัวอย่าง

P เป็นค่าความน่าจะเป็นสะสม

EBS เป็นค่าของตัวอย่างที่ได้จากการสุ่มตัวอย่างแบบใส่คืน ซึ่งอาจมีค่าซ้ำกันได้และในการวิจัยครั้งนี้ EBS คือค่าความคลาดเคลื่อนที่สุ่มได้จากความคลาดเคลื่อนที่เกิดจากการประมาณค่าตัวแปรตามด้วยตัวแบบการถดถอย หรือ $\hat{\varepsilon}_i = Y_i - \hat{Y}_i$

2. การผลิตเลขสุ่มจากการแจกแจงแบบสม่ำเสมอในช่วง (0,1)

ชุดตัวเลขที่ผลิตขึ้น(r_1, r_2, ...)ต้องมีคุณสมบัติทางสถิติที่สำคัญ 2 ประการคือ ความเป็นสม่ำเสมอ(uniform) และความเป็นอิสระ(independent) ตัวเลขสุ่ม r_i แต่ละตัวจะถูกเลือกอย่างเป็นอิสระหรือสุ่มจากเลขสุ่ม R ที่มีการแจกแจงแบบสม่ำเสมอในช่วง (0,1)

วิธีการผลิตเลขสุ่มแบบ linear congruential method เป็นวิธีการผลิตเลขสุ่มที่จะผลิตชุดตัวเลขสุ่มจำนวนเต็ม X_1, X_2, ... มีค่าระหว่าง 0 ถึง M–1 จากสมการตัวผลิต

$$X_i = (aX_{i-1} + c) \bmod M \qquad ; i = 1, 2, ...$$

เมื่อ a เป็นค่าคงที่ใดๆ

c เป็นค่าส่วนเพิ่ม(increment)

X_0 เป็นตัวเลขนำหรือค่าเริ่มต้นของการผลิตเลขสุ่ม

M เป็น modulus

mod หมายความว่า เศษที่เกิดจากการหาร (aX_{i-1} + c) ด้วย M จะเป็นเลขสุ่ม X_i และเป็นเลขสุ่มคล้ายที่จะใช้สุ่มเลขตัวต่อไป

ตัวเลขจำนวนเต็ม $X_1, X_2, \ldots$ จากสมการข้างต้นจะเป็นเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง (0, M-1) เพราะฉะนั้น ตัวเลขสุ่ม $X_1, X_2, \ldots$ ที่มีการแจกแจงแบบสม่ำเสมอในช่วง (0,1) สามารถผลิตได้จากสมการ

$$R_i = X_i / M \qquad ; i = 1, 2, \ldots.$$

ถ้ากำหนดค่า $c \neq 0$ เรียกตัวผลิตเลขสุ่มนั้นว่า mixed congruential method แต่ถ้ากำหนด $c = 0$ เรียกตัวผลิตเลขสุ่มนั้นว่า multiplicative congruential method การกำหนดค่า c, a, M และ X_0 มีความสำคัญมาก เนื่องจากมีผลโดยตรงต่อคุณสมบัติทางสถิติและความยาวของชุดตัวเลขสุ่ม จากสมการ $R_i = X_i / M$ จะได้ว่า R_i มีค่าอยู่ในเซตของ {0, 1/M, 2/M, ..., (m-1)/M} ทั้งนี้เพราะค่าของ X_i เป็นจำนวนเต็มที่อยู่ในเซตของ {0, 1, 2, ..., M-1} เพราะฉะนั้นค่า R_i จึงมีค่าไม่ต่อเนื่อง แทนที่จะเป็นค่าที่ต่อเนื่องที่มีการแจกแจงแบบสม่ำเสมอในช่วง (0,1) อย่างไรก็ตามจะประมาณความต่อเนื่องได้ โดยการกำหนดให้ M มีค่าใหญ่มากๆ จะมีผลทำให้ช่องว่าง R_i ; i = 1, 2, ... มีค่าเล็กลง ทำให้ได้ค่า R_i ที่มีความต่อเนื่องโดยประมาณ ลักษณะการกระทำดังกล่าวเป็นการสร้างความหนาแน่น(density) ในกลุ่มตัวเลขสุ่มให้มีความหนาแน่นสูงในช่วง (0,1) และเพื่อหลีกเลี่ยงชุดตัวเลขสุ่มซ้ำในการใช้งานครั้งหนึ่งๆ ตัวผลิตควรมีความยาวของชุดตัวเลขสุ่มมากที่สุดเท่าที่จะเป็นไปได้ จากการทดสอบมาแล้วเป็นจำนวนมาก วิธีการผลิตเลขสุ่มที่มีคุณสมบัติต่างๆ ดังที่กล่าวไว้ข้างต้น ก็คือวิธี multiplicative congruential ที่กำหนด $a = 7^5 = 16807$ การกำหนดค่า M ให้มีขนาดใหญ่มากๆ และเป็นเลขคี่ที่สามารถคำนวณได้จากเครื่องคอมพิวเตอร์โดยที่ $M = 2^b$ เมื่อ b เป็นค่าความยาว 1 word หรือจำนวน bit ใน 1 word ของเครื่องคอมพิวเตอร์ซึ่งเท่ากับ 32 bit โดย 1 bit สุดท้ายใช้สำหรับแสดงเครื่องหมายดังนั้นเลขจำนวนเต็มที่ใหญ่ที่สุดใน 1 word และเป็นเลขคี่ที่คอมพิวเตอร์รับได้ ก็คือ2^{b-1}-1 หรือ $2^{31}-1 = 2147483647$ นั่นคือจะได้ M = 2147483647 โปรแกรมย่อยที่ใช้ในการผลิตเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง (0,1) แสดงได้ดังนี้

```
***************************************************************************************
*          FUNCTION for generated random number Uniform(0,1)          *
***************************************************************************************

    FUNCTION RAND(IX)
    IX = IX*16807
    IF (IX.LT.0) IX=(IX+2147483647)+1
    RAND=IX
    RAND=RAND*0.4656613E-9
    RETURN
    END
```

3. โปรแกรมย่อยที่ใช้ในการสร้างข้อมูลที่มีการแจกแจงแบบปกติ

```
*************************************************************************************
*          FUNCTION for generated data from Normal Distribution                     *
*************************************************************************************
      FUNCTION NORMAL(DMEAN,SIGMA)
      REAL NORMAL,RU1,RU2,U1,U2,PI
      COMMON/SEED/IX,KN
      PI=3.142857143
      IF(KN.EQ.1)GO TO 405
      RU1=RAND(IX)
      RU2=RAND(IX)
      U1=SQRT(-2*ALOG(RU1))*COS(2*PI*RU2)
      U2=SQRT(-2*ALOG(RU1))*SIN(2*PI*RU2)
      NORMAL=DMEAN+SIGMA*U1
      KN=1
      RETURN
405   NORMAL=DMEAN+SIGMA*U2
      KN=0
      RETURN
      END
```

4. โปรแกรมที่ใช้ในการคำนวณค่าวิกฤติจากการแจกแจงแบบเอฟ

```
*****  Program for Compute Critical Value from F-Distribution  *****
       REAL PF,DF1,DF2,F
       DATA AF,BF,XF,IER/0.0,0.0,0.0,0/
       READ(*,*)PF,DF1,DF2
       AF=0.0
       BF=0.0
       AF=DF1/2
       BF=DF2/2
       CALL MDBETI(PF,AF,BF,XF,IER)
       F=(DF2*XF)/(DF1-DF1*XF)
       WRITE(*,*)PF,DF1,DF2,F
       STOP
       END
***********************************************************************************************
* SUBROUTINE   MDBETI(PF,AF,BF,IER)                                                           *
* FUNCTION INVERSE INCOMPLETE BETA PROBABILITY DISTRIBUTION FUNCTION                          *
* PARAMETER  PF-INPUT:PROBABILITY IN THE EXCLUSIVE RANGE                                      *
*                AF-INPUT:FIRST PARAMETER OF THE INCOMPLETE BETA PDF                          *
*                BF-INPUT:SECOND PARAMETER OF THE INCOMPLETE BETA PDF                         *
*              XF-OUTPUT:VALUE SUCH THAT THE PROBABILITY THAT A RANDOM                        *
*                VARIABLE DISTRIBUTED BETA(AF,BF) IS LESS THAN OR EQUAL P.                    *
***********************************************************************************************
       SUBROUTINE MDBETI(PF,AF,BF,XF,IER)
       DATA EPS,SIG/.0001,1.E-5/
       DATA ZERO,ITMAX/0.,30/
       IER=0
       IC=0
       AB=AF/BF
       XLF=0.0
       XRF=1.0
```

```
      FXL=-PF
      FXR=1.0-PF
      IF(FXL*FXR.GT.ZERO)GO TO 25
5     XF=(XLF+XRF)*.5
      CALL MDBETA(XF,AF,BF,P1,IER)
      IF(IER.NE.0)GO TO 20
      FCS=P1-PF
      IF(FCS*FXL.GT.ZERO)GO TO 10
      XRF=XF
      FXR=FCS
      GO TO 15
10    XLF=XF
      FXL=FCS
15    XRMXL=XRF-XLF
      IF(XRMXL.LE.SIG.AND.ABS(FCS).LE.EPS)GO TO 9005
      IC=IC+1
      IF(IC.LE.ITMAX)GO TO 5
      IER=130
      GO TO 9000
20    IER=129
      GO TO 9000
25    IER=131
9000  CONTINUE
9005  RETURN
      END
      SUBROUTINE MDBETA(XF,AF,BF,PF,IER)
      DOUBLE PRECISION PS,PX,Y,P1,DA,XINT,CNT,WH,XB,DB,C
     *,EPS,EPS1,ALEPS,TOT,PQ,D4,DD,PA
      DATA EPS,EPS1,ALEPS/1.D-6,1.D-78,-179.6016D0/
      Y=XF
      IF((XF.LE.1.0).AND.(XF.GE.0.0)) GO TO 5
```

```
      IER=129
      GO TO 9000
5     IF((AF.GT.0.0).AND.(BF.GT.0.0)) GO TO 10
      IER=130
      GO TO 9000
10    IER=0
      AA=AF
      BB=BF
      IF(XF.GT.0.5) GO TO 15
      INT=0
      GO TO 20
15    INT=1
      TEMP=AA
      AA=BB
      BB=TEMP
      Y=1.D0-Y
20    IF(XF.NE.0.0.AND.XF.NE.1.0)GO TO 25
      PF=0.
      GO TO 60
25    IB=BB
      TEMP=IB
      PS=BB-FLOAT(IB)
      IF(BB.EQ.TEMP)PS=1.D0
      DA=AA
      DB=BB
      PX=DA*DLOG(Y)
      DD=DA+DB
      PQ=GAMMLN(DD)
      P1=GAMMLN(DA)
      C=GAMMLN(DB)
      D4=DLOG(DA)
```

```
      PA=PS+DA
      XB=PX+GAMMLN(PA)-GAMMLN(PS)-D4-P1
      IB=XB/ALEPS
      XINT=0.D0
      IF(IB.NE.0)GO TO 35
      XINT=DEXP(XB)
      CNT=XINT*DA
      WH=0.0D0
30    WH=WH+1.D0
      CNT=CNT*(WH-PS)*Y/WH
      XB=CNT/(DA+WH)
      XINT=XINT+XB
      IF(XB/EPS.GT.XINT) GO TO 30
35    TOT=0.D0
      IF(DB.LE.1.D0)GO TO 55
      XB=PX+DB*DLOG(1.D0-Y)+PQ-P1-DLOG(DB)-C
      IB=XB/ALEPS
      IF(IB.LE.0)IB=0
      C=1.D0/(1.D0-Y)
      CNT=DEXP(XB-DFLOAT(IB)*ALEPS)
      PS=DB
      WH=DB
40    WH=WH-1.D0
      IF(WH.LE.0.0D0)GO TO 55
      PX=(PS*C)/(DA+WH)
      IF(PX.GT.1.D0)GO TO 45
      IF(CNT/EPS.LE.TOT.OR.CNT.LE.EPS1/PX)GO TO 55
45    CNT=CNT*PX
      IF(CNT.LE.1.00) GO TO 50
      IB=IB-1
      CNT=CNT*EPS1
```

```
50    PS=WH
      IF(IB.EQ.0)TOT=TOT+CNT
      GO TO 40
55    PF=TOT+XINT
60    IF(INT.NE.0)PF=1.-PF
      GO TO 9005
9000  CONTINUE
9005  RETURN
      END
      FUNCTION GAMMLN(XX)
      DOUBLE PRECISION SER,STP,TMP,X,Y,COF(6),XX
      DATA COF,STP/76.18009172947146D0,-86.50532032941677D0
     *,24.01409824083091D0,-1.231739572450155D0,.1208650973866179D-2
     *,-.5395239384953D-5,2.5066282746310005D0/
      X=XX
      Y=X
      TMP=X+5.5D0
      TMP=(X+0.5D0)*LOG(TMP)-TMP
      SER=1.000000000190015D0
      DO 10 J=1,6
      Y=Y+1.D0
      SER=SER+COF(J)/Y
10    CONTINUE
      GAMMLN=TMP+LOG(STP*SER/X)
      RETURN
      END
```

ภาคผนวก ข

ตารางที่ ข.1 ตัวอย่างผลการวิเคราะห์ เมื่อกำหนดค่าสัมประสิทธิ์การถดถอยที่ระดับต่าง ๆ โดยพิจารณาค่าอำนาจการทดสอบ ที่ระดับนัยสำคัญ 0.05 และ $\varepsilon_i \sim N(0,1)$ ในตัวแบบที่ 2 ($\beta_2 = 3$)

n	β_0	β_1	ตัวสถิติทดสอบ		
			F	KS	CvM
10	0	-5	0.448	0.256	0.241
		-1	0.456	0.238	0.250
		1	0.454	0.225	0.233
		5	0.442	0.245	0.250
	1	-5	0.454	0.264	0.247
		-1	0.465	0.250	0.269
		1	0.459	0.235	0.250
		5	0.471	0.251	0.254
	5	-5	0.462	0.258	0.259
		-1	0.481	0.260	0.272
		1	0.468	0.247	0.265
		5	0.479	0.257	0.271
30	0	-5	0.505	0.404	0.416
		-1	0.514	0.405	0.408
		1	0.520	0.410	0.415
		5	0.508	0.409	0.412
	1	-5	0.510	0.410	0.415
		-1	0.516	0.407	0.414
		1	0.526	0.411	0.420
		5	0.515	0.414	0.422
	5	-5	0.524	0.418	0.433
		-1	0.533	0.420	0.422
		1	0.535	0.419	0.422
		5	0.522	0.425	0.431

ตารางที่ ข.2 ตัวอย่างผลการวิเคราะห์ เมื่อกำหนดจำนวนระดับของตัวแปรอิสระต่าง ๆ โดยพิจารณาค่าอำนาจการทดสอบ ที่ระดับนัยสำคัญ 0.05 และ $\varepsilon_i \sim N(0,1)$ ในตัวแบบที่ 2 ($\beta_2 = 3$)

n	k	ตัวสถิติทดสอบ		
		F	KS	CvM
10	5	0.467	0.220	0.241
	6	0.464	0.224	0.240
	7	0.460	0.222	0.245
20	5	0.488	0.266	0.264
	6	0.485	0.268	0.260
	7	0.486	0.265	0.267
30	5	0.500	0.401	0.357
	6	0.504	0.405	0.358
	7	0.505	0.404	0.360
50	5	0.543	0.455	0.453
	6	0.541	0.458	0.450
	7	0.540	0.457	0.452
70	5	0.557	0.459	0.466
	6	0.559	0.460	0.461
	7	0.599	0.465	0.462
100	5	0.614	0.587	0.565
	6	0.611	0.589	0.568
	7	0.616	0.590	0.577

จากตารางที่ ข.1 และตารางที่ ข.2 จะได้ว่าไม่ว่าจะกำหนดให้ค่าสัมประสิทธิ์การถดถอยหรือจำนวนระดับของตัวแปรอิสระมีค่าเท่าใด ผลการวิเคราะห์ที่ได้จากการทดสอบเทียบความกลมกลืนสำหรับตัวแบบการถดถอยจะสอดคล้องกัน กล่าวคือเมื่อพิจารณาจากค่าอำนาจการทดสอบในตารางข้างต้นแล้ว ตัวสถิติเอฟจะเหมาะสมสำหรับการทดสอบนี้จากสถานการณ์ที่กำหนดในทุกค่าของค่าสัมประสิทธิ์การถดถอยหรือจำนวนระดับของตัวแปรอิสระ

ตารางที่ ข.3 ตัวอย่างค่าวิกฤติจากวิธีการแบบบูสแตรป สำหรับตัวสถิติทดสอบ Kolmogorov-Smirnov และตัวสถิติทดสอบ Cramer-von Mises เมื่อพิจารณาตัวแบบที่ 1 ที่ $\varepsilon_i \sim N(0,1)$

n	B	สถิติทดสอบ Kolmogorov-Smirnov			ตัวสถิติทดสอบ Cramer-von Mises		
		90%	95%	99%	90%	95%	99%
10	300	0.4994	0.5516	0.6492	0.2463	0.2826	0.3497
	400	0.5064	0.5524	0.6398	0.2513	0.2826	0.3555
	500	0.5125	0.5597	0.6495	0.2567	0.2934	0.3724
	600	0.5064	0.5519	0.6495	0.2513	0.2913	0.3768
	700	0.5125	0.5598	0.6497	0.2553	0.2974	0.3775
	800	0.5107	0.5580	0.6547	0.2543	0.2972	0.3766
	1000	0.5047	0.5597	0.6586	0.2481	0.2958	0.3736
20	300	0.5631	0.6231	0.7222	0.4508	0.6066	1.1604
	400	0.5891	0.6613	0.7405	0.4713	0.6357	1.1516
	500	0.5833	0.6541	0.7822	0.4686	0.6009	1.0409
	600	0.5848	0.6348	0.7827	0.4771	0.6376	1.0545
	700	0.5884	0.6355	0.7542	0.4724	0.6155	1.0379
	800	0.5831	0.6799	0.7968	0.4871	0.6134	1.0392
	1000	0.5935	0.6790	0.7353	0.4922	0.6277	1.0407
30	300	0.5258	0.6089	0.6713	0.2529	0.2993	0.4505
	400	0.5655	0.7492	0.9638	0.4018	0.5246	1.0161
	500	0.5524	0.8654	1.0306	0.2009	0.6467	0.8343
	600	0.5431	0.8597	1.0419	0.2008	0.6527	0.8393
	700	0.5568	0.8558	0.9952	0.1996	0.6334	0.8322
	800	0.5499	0.8672	0.9829	0.2072	0.6446	0.8307
	1000	0.5554	0.8564	1.0856	0.1987	0.6445	0.8342

จากตารางข้างต้นจะเห็นว่า ที่จำนวนรอบการสุ่มตัวอย่างแบบบูตสแตรปเท่ากับ 500 รอบนั้น จะได้ค่าวิกฤติที่เริ่มคงที่ กล่าวคือ เมื่อจำนวนรอบมากขึ้นก็ให้ค่าวิกฤติที่มีค่าแตกต่างกันเพียงเล็กน้อย ดังนั้น ผู้วิจัยจึงกำหนดให้การวิจัยครั้งนี้ทำการสุ่มตัวอย่างแบบบูตสแตรปจำนวน 500 รอบก็เพียงพอแล้ว

ตารางที่ ข.4 ตัวอย่างการวิเคราะห์ เมื่อกำหนดค่าเฉลี่ยและค่าความแปรปรวนของตัวแปรอิสระที่ระดับต่างๆ **พิจารณาค่าความผิดพลาดประเภทที่ 1** ที่ระดับนัยสำคัญ 0.05 และ $\varepsilon_i \sim N(0,1)$

n	(μ,σ^2)	α	สถิติทดสอบ		
			F	KS	CvM
20	(20,4)	0.01	0.008	0.007	0.009
		0.05	0.047	0.046	0.055
		0.10	0.105	0.116	0.123
	(20,36)	0.01	0.008	0.006	0.008
		0.05	0.045	0.049	0.056
		0.10	0.092	0.121	0.122
	(20,100)	0.01	0.010	0.008	0.009
		0.05	0.045	0.049	0.054
		0.10	0.101	0.117	0.119
	(20,196)	0.01	0.009	0.008	0.009
		0.05	0.050	0.047	0.053
		0.10	0.104	0.129	0.121
30	(20,4)	0.01	0.016	0.008	0.013
		0.05	0.063	0.054	0.054
		0.10	0.102	0.114	0.105
	(20,36)	0.01	0.013	0.011	0.012
		0.05	0.052	0.056	0.051
		0.10	0.094	0.109	0.116
	(20,100)	0.01	0.011	0.004	0.013
		0.05	0.049	0.050	0.057
		0.10	0.103	0.102	0.102
	(20,196)	0.01	0.015	0.008	0.002
		0.05	0.062	0.055	0.028
		0.10	0.070	0.109	0.094

ตารางที่ ข.5 ตัวอย่างการวิเคราะห์ เมื่อกำหนดค่าเฉลี่ยและค่าความแปรปรวนของตัวแปรอิสระที่ระดับต่างๆ พิจารณาค่าอำนาจการทดสอบ ที่ระดับนัยสำคัญ 0.05 และ $\varepsilon_i \sim N(0,1)$ ในตัวแบบที่ 2 ($\beta_2 = 3$)

n	(μ,σ^2)	α	สถิติทดสอบ		
			F	KS	CvM
20	(20,4)	0.01	0.451	0.322	0.324
		0.05	0.477	0.345	0.330
		0.10	0.502	0.387	0.381
	(20,36)	0.01	0.455	0.330	0.320
		0.05	0.470	0.352	0.354
		0.10	0.498	0.385	0.377
	(20,100)	0.01	0.450	0.320	0.323
		0.05	0.485	0.342	0.350
		0.10	0.522	0.380	0.380
	(20,196)	0.01	0.450	0.329	0.325
		0.05	0.482	0.351	0.355
		0.10	0.560	0.384	0.381
30	(20,4)	0.01	0.471	0.366	0.365
		0.05	0.495	0.387	0.384
		0.10	0.533	0.412	0.420
	(20,36)	0.01	0.479	0.370	0.377
		0.05	0.501	0.385	0.389
		0.10	0.540	0.420	0.421
	(20,100)	0.01	0.474	0.367	0.365
		0.05	0.505	0.384	0.388
		0.10	0.556	0.411	0.417
	(20,196)	0.01	0.477	0.359	0.360
		0.05	0.500	0.385	0.384
		0.10	0.549	0.423	0.420

จากตารางที่ ข.4 และ ตารางที่ ข.5 ข้างต้น จะได้ว่าที่ขนาดตัวอย่างเดียวกัน ค่าประมาณความผิดพลาดประเภทที่ 1 จะมีค่าใกล้เคียงกัน และเมื่อพิจารณาค่าอำนาจการทดสอบของตัวสถิติทดสอบทั้งสาม จะได้ว่าตัวสถิติทดสอบที่เหมาะสมสำหรับการทดสอบเทียบความกลมกลืนในสถานการณ์ตามที่กำหนดนั้นจะเป็นตัวสถิติทดสอบตัวเดียวกัน แม้ว่าค่าเฉลี่ยและค่าความแปรปรวนของตัวแปรอิสระจะเปลี่ยนไปก็ตาม ดังนั้นผู้วิจัยจึงกำหนดให้ค่าเฉลี่ยและค่าความแปรปรวนของตัวแปรอิสระเป็นค่าใดๆ ดังที่กำหนดไว้ในแผนการทดลอง

ภาคผนวก ค

โปรแกรมที่ใช้ในการวิจัย

1. กรณีข้อมูลมีค่าซ้ำกัน

```
****************************************************************************
*****   REPEAT OBSERVATIONS -  CLASSICAL BOOTSTRAP      *
****************************************************************************
     REAL Y(100),E(100),X(100,10),B(10),XL(100),BT(10)
    *,RN(100),RN1(100),RN2(100),PP(100),YI(100),EE(100)
    *,MSLF,MSPE,YE(100),MEANY
    *,P(100),NORMAL,FN(100),FNX(100),XMEAN(20),XSIGMA(20)
     REAL D01,D05,D10,W01,W05,W10
    *    ,SD01,SD05,SD10,SW01,SW05,SW10
     COMMON/SEED/IX,KN
    *    /SAMPLE/X,E
    *    /PROB/P
    *    /BETA/BT
     OPEN(3,FILE='OUTP6.DAT',STATUS='UNKNOWN')
     OPEN(4,FILE='INP5.DAT',STATUS='OLD')
     IX=335687
     NOFF=0
7    READ(4,*)MODEL,N,NP,KE
     WRITE(3,*)MODEL,N,NP,KE
     IF((MODEL.EQ.1).OR.(MODEL.EQ.2))THEN
       M=MODEL+1
       READ(4,*)XMEAN(2),XSIGMA(2)
       WRITE(3,*)'MEAN AND SIGMA OF X(1)',XMEAN(2),XSIGMA(2)
     ELSE IF((MODEL.EQ.3).OR.(MODEL.EQ.4))THEN
       M=MODEL
       READ(4,*) XMEAN(2),XSIGMA(2),XMEAN(3),XSIGMA(3)
       WRITE(3,*)XMEAN(2),XSIGMA(2),XMEAN(3),XSIGMA(3)
     END IF
     READ(4,*) (BT(I),I=1,M)
     WRITE(3,*)'REAL BETA',(BT(I),I=1,M)
     KN=0
     NI=0
```

```
      NN=NP
      SD01=0.0
      SD05=0.0
      SD10=0.0
      SW01=0.0
      SW05=0.0
      SW10=0.0
      SF01=0.0
      SF05=0.0
      SF10=0.0
      D01=0.0
      D05=0.0
      D10=0.0
      W01=0.0
      W05=0.0
      W10=0.0
5     DO 1 I=1,NN
       PP(I)=FLOAT(I)/FLOAT(NN)
       X(I,1)=1.0
       DO 3 II=2,M
       DMEAN=XMEAN(II)
       SIGMA=XSIGMA(II)
       X(I,II)=NORMAL(DMEAN,SIGMA)
       IF((II+1.EQ.M).AND.(MODEL.EQ.2))THEN
        X(I,3)=X(I,2)*X(I,2)
        GO TO 1
       ELSE IF((II+1.EQ.M).AND.(MODEL.EQ.4))THEN
        X(I,4)=X(I,2)*X(I,3)
        GO TO 1
       END IF
3      CONTINUE
1     CONTINUE
      DO 4 I=NN+1,N
      CR=RAND(IX)
      DO 6 J=1,NN
```

```
      JJ=J-1
      IF(JJ.EQ.0)THEN
       G1=0.0
      ELSE
       G1=PP(JJ)
      END IF
      G2=PP(J)
      IF((CR.GT.G1).AND.(CR.LE.G2))THEN
      DO 8 K=1,M
       X(I,K)=X(J,K)
8     CONTINUE
      GO TO 4
      END IF
6     CONTINUE
4     CONTINUE
      DO 10 I=1,N
      Y(I)=0.0
      A=0.0
      DO 12 J=1,M
       A=A+X(I,J)*BT(J)
12    CONTINUE
      IF (KE.EQ.1)THEN
       EMEAN=0.0
       ESIGMA=1.0
       E(I)=NORMAL(EMEAN,ESIGMA)
       Y(I)=A+E(I)
      ELSE IF (KE.EQ.2)THEN
       EMEAN=0.0
       ESIGMA=SQRT(2.0)
       E(I)=NORMAL(EMEAN,ESIGMA)
       Y(I)=A+E(I)
      ELSE IF (KE.EQ.3)THEN
       EMEAN=0.0
       ESIGMA=SQRT(3.0)
       E(I)=NORMAL(EMEAN,ESIGMA)
```

```
Y(I)=A+E(I)
ELSE IF (KE.EQ.4)THEN
EMEAN=0.0
ESIGMA=0.5
A1=NORMAL(EMEAN,ESIGMA)
E(I)=EXP(A1)
Y(I)=A+E(I)
ELSE IF (KE.EQ.5)THEN
EMEAN=0.0
ESIGMA=1.0
A1=NORMAL(EMEAN,ESIGMA)
E(I)=EXP(A1)
Y(I)=A+E(I)
ELSE IF (KE.EQ.6)THEN
EMEAN=0.0
ESIGMA=1.5
A1=NORMAL(EMEAN,ESIGMA)
E(I)=EXP(A1)
Y(I)=A+E(I)
ELSE IF (KE.EQ.7) THEN
ALP=2.0
BTA=1.0
E(I)=GAMMA1(ALP,BTA)
Y=A+E(I)
ELSE IF (KE.EQ.8) THEN
ALP=3.0
BTA=1.0
E(I)=GAMMA1(ALP,BTA)
Y=A+E(I)
ELSE IF (KE.EQ.9) THEN
ALP=4.0
BTA=1.0
E(I)=GAMMA1(ALP,BTA)
Y=A+E(I)
END IF
```

```
      P(I)=FLOAT(I)/FLOAT(N)
 10   CONTINUE
******************************************************************************
*                              F-Test                                        *
******************************************************************************
      CALL OLS(X,Y,B,M,N,SS1,YY,SUMY,YE)
      DO 11 IE=1,N
      EE(IE)=Y(IE)-YE(IE)
 11   CONTINUE
      L=0
      SSPE=0.0
      DO 50 I=1,N
       K=0
       SUMYI=0.0
       YBARI=0.0
       SSYI=0.0
       DO 55 II=1,N
       IF (X(I,2).NE.X(II,2)) GO TO 55
       SUMYI=SUMYI+Y(II)
       K=K+1
       YI(K)=Y(II)
55     CONTINUE
      IF (I.EQ.1) GO TO 62
      DO 60 J=1,I-1
        IF (X(I,2).EQ.X(J,2)) THEN
          GO TO 50
        END IF
60    CONTINUE
62    YBARI=SUMYI/K
      DO 65 KK=1,K
        SSYI=SSYI+(YI(KK)-YBARI)**2
65    CONTINUE
      SSPE=SSPE+SSYI
      L=L+1
      XL(L)=X(I,2)
```

```
      FN(L)=FLOAT(K)/FLOAT(N)
 50   CONTINUE
      MEANY=SUMY/FLOAT(N)
      SSR=SS1-(FLOAT(N)*(MEANY**2))
      SSTO=YY-(FLOAT(N)*(MEANY**2))
      DFR=FLOAT(M)-1.
      DFE=FLOAT(N)-DFR-1.
      DFLF=FLOAT(L)-DFR-1.
      DFPE=FLOAT(N)-FLOAT(L)
      SSE=SSTO-SSR
      SSLF=SSE-SSPE
      MSLF=SSLF/DFLF
      MSPE=SSPE/DFPE
      FCAL=MSLF/MSPE
      AF=DFLF/2.
      BF=DFPE/2.
      PF=0.99
      CALL MDBETI(PF,AF,BF,XF,IER)
      F01=(DFPE*XF)/(DFLF-DFLF*XF)
      PF=0.95
      CALL MDBETI(PF,AF,BF,XF,IER)
      F05=(DFPE*XF)/(DFLF-DFLF*XF)
      PF=0.90
      CALL MDBETI(PF,AF,BF,XF,IER)
      F10=(DFPE*XF)/(DFLF-DFLF*XF)
      IF (FCAL.GT.F01) SF01=SF01+1.0
      IF (FCAL.GT.F05) SF05=SF05+1.0
      IF (FCAL.GT.F10) SF10=SF10+1.0
*********************************************************************************
*                    Kolmogorov-Smirnov Test                                    *
*********************************************************************************
      CALL RNX(L,M,N,X,Y,B,XL,RN,RN1,RN2)
      CALL MAXDT(SUP,RN1,L)
      DCAL=SUP
      DO 68 I=1,L
```

```
      FNX(I)=0.0
      DO 67 J=1,L
      IF (XL(J).LE.XL(I)) THEN
      FNX(I)=FNX(I)+FN(J)
      END IF
 67   CONTINUE
 68   CONTINUE
      CALL BOOTS(EE,FNX,XL,L,M,N,D01,D05,D10,W01,W05,W10)
      IF (DCAL.GT.D01) SD01=SD01+1.0
      IF (DCAL.GT.D05) SD05=SD05+1.0
      IF (DCAL.GT.D10) SD10=SD10+1.0
**********************************************************************
*                    Cramer-von Mises Test                           *
**********************************************************************
      WCAL=0.0
      DO 70 I=1,L
      WCAL=WCAL+(RN2(I)*FNX(I))
 70   CONTINUE
      IF (WCAL.GT.W01) SW01=SW01+1.0
      IF (WCAL.GT.W05) SW05=SW05+1.0
      IF (WCAL.GT.W10) SW10=SW10+1.0
**********************************************************************
*              Type I Error  and Power of the Test                   *
**********************************************************************
      NI=NI+1
      IF (NI.LT.1000) GO TO 5
      PF01=SF01/1000.
      PF05=SF05/1000.
      PF10=SF10/1000.
      PD01=SD01/1000.
      PD05=SD05/1000.
      PD10=SD10/1000.
      PW01=SW01/1000.
      PW05=SW05/1000.
      PW10=SW10/1000.
```

```
      WRITE(3,*) '       F-TEST        KS-TEST       CvM-TEST'
      WRITE(3,*) '99%  ',PF01,PD01,PW01
      WRITE(3,*) '95%  ',PF05,PD05,PW05
      WRITE(3,*) '90%  ',PF10,PD10,PW10

*          Test Significant for Type I Error                          *
      IF((MODEL.EQ.1).OR.(MODEL.EQ.3))THEN
      Z01=0.01+1.645*SQRT(0.0099/FLOAT(NI))
      Z05=0.05+1.645*SQRT(0.0475/FLOAT(NI))
      Z10=0.10+1.645*SQRT(0.09/FLOAT(NI))
      IF((PF01.GE.0.0).AND.(PF01.LE.Z01))THEN
       WRITE(3,*)'F-test Controlable type I Error at 0.01'
      END IF
      IF((PF05.GE.0.0).AND.(PF05.LE.Z05))THEN
       WRITE(3,*)'F-test Controlable type I Error at 0.05'
       END IF
      IF((PF10.GE.0.0).AND.(PF10.LE.Z10))THEN
       WRITE(3,*)'F-test Controlable type I Error at 0.10'
      END IF
      IF((PD01.GE.0.0).AND.(PD01.LE.Z01))THEN
       WRITE(3,*)'KS-test Controlable type I Error at 0.01'
      END IF
      IF((PD05.GE.0.0).AND.(PD05.LE.Z05))THEN
       WRITE(3,*)'KS-test Controlable type I Error at 0.05'
      END IF
      IF((PD10.GE.0.0).AND.(PD10.LE.Z10))THEN
       WRITE(3,*)'KS-test Controlable type I Error at 0.10'
      END IF
      IF((PW01.GE.0.0).AND.(PW01.LE.Z01))THEN
       WRITE(3,*)'CVM-test Controlable type I Error at 0.01'
      END IF
      IF((PW05.GE.0.0).AND.(PW05.LE.Z05))THEN
       WRITE(3,*)'CVM-test Controlable type I Error at 0.05'
      END IF
      IF((PW10.GE.0.0).AND.(PW10.LE.Z10))THEN
```

```
      WRITE(3,*)'CVM-test Controlable type I Error at 0.10'
      END IF
      END IF
      NOFF=NOFF+1
      IF(NOFF.LT.42)GO TO 7
      CLOSE(3)
      CLOSE(4)
      STOP
      END
*********************************************************************************
*                    Subroutine for Estimated  Rn(x)                            *
*********************************************************************************
      SUBROUTINE RNX(L,M,N,X,Y,B,XL,RN,RN1,RN2)
      DIMENSION X(100,10),Y(100),YH(100),B(10),XL(100)
     *      ,R1(100),RN2(100),RN1(100),RN(100)
      DO 200 K=1,L
      RN(K)=0.0
      RN1(K)=0.0
      RN2(K)=0.0
      R1(K)=0.0
      DO 210 I=1,N
      YH(I)=0.0
      DO 220 J=1,M
      YH(I)=YH(I)+X(I,J)*B(J)
  220 CONTINUE
      IF (X(I,2).LE.XL(K)) THEN
        R1(K)=R1(K)+(Y(I)-YH(I))
      ELSE
      END IF
  210 CONTINUE
      A=FLOAT(N)
      C=1.0/SQRT(A)
      RN(K)=C*R1(K)
      RN1(K)=ABS(RN(K))
      RN2(K)=RN(K)*RN(K)
```

```
 200  CONTINUE
      RETURN
      END
*******************************************************************************
*                     Subroutine for OLS Estimater                            *
*******************************************************************************
      SUBROUTINE OLS(X,Y,B,M,N,SS1,YY,SUMY,YE)
      DIMENSION X(100,10),Y(100),B(10),XY(10),XX(10,10)
     *        ,XXI(10,10),YE(100)
      DO 20 I=1,M
      DO 22 J=1,M
       XY(J)=0.0
       XX(I,J)=0.0
       YY=0.0
       SUMY=0.0
      DO 25 K=1,N
       XX(I,J)=XX(I,J)+X(K,I)*X(K,J)
       XY(J)=XY(J)+X(K,J)*Y(K)
       YY=YY+Y(K)*Y(K)
       SUMY=SUMY+Y(K)
 25   CONTINUE
 22   CONTINUE
 20   CONTINUE
      CALL INV(XX,XXI,M)
      DO 30 I=1,M
       B(I)=0.0
      DO 31 J=1,M
       B(I)=B(I)+XXI(I,J)*XY(J)
 31   CONTINUE
 30   CONTINUE
      DO 35 I=1,N
       YE(I)=0.0
       SS1=0.0
       DO 40 J=1,M
        YE(I)=YE(I)+X(I,J)*B(J)
```

```
      SS1=SS1+B(J)*XY(J)
 40   CONTINUE
 35   CONTINUE
      RETURN
      END
*****************************************************************************
*                  Subroutine for Ranking Data                              *
*****************************************************************************
      SUBROUTINE RANK(R)
      DIMENSION R(500)
 290  IC=0
      DO 295 I=2,500
      II=I-1
      IF (R(II).LE.R(I)) GO TO 295
      Z=R(II)
      R(II)=R(I)
      R(I)=Z
      IC=1
 295  CONTINUE
      IF (IC.EQ.1) GO TO 290
      RETURN
      END
*****************************************************************************
*              Subroutine for Sampling with Replacement                     *
*****************************************************************************
      SUBROUTINE SWR(N,EBS,EE)
      DIMENSION P(100),EBS(100),EE(100)
      COMMON/PROB/P/SEED/IX,KN
      DO 300 J=1,N
      CRN=RAND(IX)
      DO 305 I=1,N
      II=I-1
      IF (II.EQ.0) THEN
       X1=0.0
      ELSE
```

```
     X1=P(II)
    END IF
     X2=P(I)
    IF((CRN.GT.X1).AND.(CRN.LE.X2)) THEN
     EBS(J)=EE(I)
     GOTO 300
    END IF
 305  CONTINUE
 300  CONTINUE
    RETURN
    END
*********************************************************************************
*       FUNCTION for generated random number Uniform(0,1)                       *
*********************************************************************************
    FUNCTION RAND(IX)
    IX = IX*16807
    IF (IX.LT.0) IX=(IX+2147483647)+1
    RAND=IX
    RAND=RAND*0.4656613E-9
    RETURN
    END
*********************************************************************************
*                       Subroutine for Inverse  Matrix                          *
*********************************************************************************
    SUBROUTINE INV(A,XXI,M)
    DIMENSION A(10,10),XXI(10,10)
    N=2*M
    N1=M+1
    M1=M-1
    DO 320 I=1,M
      M1=M1+1
      DO 320 J=N1,N
        M2=J-M1
        IF (M2.EQ.1) A(I,J)=1.0
        IF (M2.NE.1) A(I,J)=0.0
```

```
320  CONTINUE
     DO 325 I=1,M
       DO 330 K=I,M
         IF(A(K,I).EQ.0.0) GO TO 330
         I1=K
         GO TO 340
330  CONTINUE
340  IF(I1.EQ.I) GO TO 345
     DO 347 J=1,N
       E=A(I1,J)
       F=A(I,J)
       A(I,J)=E
       A(I1,J)=F
347  CONTINUE
345  D=A(I,I)
     DO 350 J=I,N
       A(I,J)=A(I,J)/D
350  CONTINUE
     DO 355 K=1,M
       IF(K.EQ.I) GO TO 355
       IF(A(K,I).EQ.0.0) GO TO 355
       C=A(K,I)
     DO 360 J=1,N
       A(K,J)=A(K,J)-(C*A(I,J))
360  CONTINUE
355  CONTINUE
325  CONTINUE
     DO 365 I=1,M
     DO 370 J=1,N
       K=J+M
       XXI(I,J)=A(I,K)
370  CONTINUE
365  CONTINUE
     RETURN
     END
```

```
******************************************************************************
*                    Subroutine for Maximized Data                           *
******************************************************************************
      SUBROUTINE MAXDT(SUP,R,L)
      DIMENSION R(100)
      SUP=R(1)
      DO 372 I=2,L
       IF (SUP.GE.R(I)) GO TO 372
       SUP=R(I)
372   CONTINUE
      RETURN
      END
******************************************************************************
*                 Subroutine for Bootstrap Approximation                     *
******************************************************************************
      SUBROUTINE BOOTS(EE,FNX,XL,L,M,N,D01,D05,D10
     *            ,W01,W05,W10)
      DIMENSION EBS(100),YBS(100),BB(10),X(100,10),FNX(100)
     *      ,RN1(100),RN2(100),DB(500),WB(500),BT(10)
     *      ,E(100),EE(100),P(100),XL(100),RN(100)
      REAL SUMY,YY,SUP,D01,D05,D10,W01,W05,W10
      COMMON/SAMPLE/X,E/PROB/P/BETA/BT/SEED/IX,KN
      D01=0.0
      D05=0.0
      D10=0.0
      W01=0.0
      W05=0.0
      W10=0.0
      DO 400 IB=1,500
      DB(IB)=0.0
      WB(IB)=0.0
      CALL SWR(N,EBS,EE)
      DO 375 I=1,N
      YBS(I)=0.0
      DO 377 J=1,M
```

```
    YBS(I)=YBS(I)+(X(I,J)*BT(J))
377 CONTINUE
    YBS(I)=YBS(I)+EBS(I)
375 CONTINUE
    CALL OLS(X,YBS,BB,M,N,SS1,YY,SUMY,YE)
    CALL RNX(L,M,N,X,YBS,BB,XL,RN,RN1,RN2)
    CALL MAXDT(SUP,RN1,L)
    DB(IB)=SUP
    DO 380 I=1,L
    WB(IB)=WB(IB)+(RN2(I)*FNX(I))
380 CONTINUE
400 CONTINUE
    CALL RANK(DB)
    D01=DB(495)
    D05=DB(475)
    D10=DB(450)
    CALL RANK(WB)
    W01=WB(495)
    W05=WB(475)
    W10=WB(450)
    RETURN
    END
************************************************************************
*       FUNCTION for generate data from Normal Distribution           *
************************************************************************
    FUNCTION NORMAL(DMEAN,SIGMA)
    REAL NORMAL,RU1,RU2,U1,U2,PI
    COMMON/SEED/IX,KN
    PI=3.142857143
    IF(KN.EQ.1)GO TO 405
    RU1=RAND(IX)
    RU2=RAND(IX)
    U1=SQRT(-2*ALOG(RU1))*COS(2*PI*RU2)
    U2=SQRT(-2*ALOG(RU1))*SIN(2*PI*RU2)
    NORMAL=DMEAN+SIGMA*U1
```

```
      KN=1
      RETURN
405   NORMAL=DMEAN+SIGMA*U2
      KN=0
      RETURN
      END
**********************************************************************************************************
* SUBROUTINE   MDBETI(PF,AF,BF,IER)                                                                      *
* FUNCTION     -INVERSE INCOMPLETE BETA PROBABILITY DISTRIBUTION FUNCTION                                *
* PARAMETER  PF-INPUT:PROBABILITY IN THE EXCLUSIVE RANGE                                                 *
*            AF-INPUT:FIRST PARAMETER OF THE INCOMPLETE BETA PDF                                         *
*            BF-INPUT:SECOND PARAMETER OF THE INCOMPLETE BETA PDF                                        *
*            XF-OUTPUT:VALUE SUCH THAT THE PROBABILITY THAT A RANDOM VARIABLE                            *
*              DISTRIBUTED BETA(AF,BF) IS LESS THAN OR EQUAL P.                                          *
**********************************************************************************************************
      SUBROUTINE MDBETI(PF,AF,BF,XF,IER)
      DATA EPS,SIG/.0001,1.E-5/
      DATA ZERO,ITMAX/0.,30/
      IER=0
      IC=0
      AB=AF/BF
      XLF=0.0
      XRF=1.0
      FXL=-PF
      FXR=1.0-PF
      IF(FXL*FXR.GT.ZERO)GO TO 25
5     XF=(XLF+XRF)*.5
      CALL MDBETA(XF,AF,BF,P1,IER)
      IF(IER.NE.0)GO TO 20
      FCS=P1-PF
      IF(FCS*FXL.GT.ZERO)GO TO 10
      XRF=XF
      FXR=FCS
      GO TO 15
10    XLF=XF
```

```
      FXL=FCS
 15   XRMXL=XRF-XLF
      IF(XRMXL.LE.SIG.AND.ABS(FCS).LE.EPS)GO TO 9005
      IC=IC+1
      IF(IC.LE.ITMAX)GO TO 5
      IER=130
      GO TO 9000
 20   IER=129
      GO TO 9000
 25   IER=131
 9000 CONTINUE
 9005 RETURN
      END
      SUBROUTINE MDBETA(XF,AF,BF,PF,IER)
      DOUBLE PRECISION PS,PX,Y,P1,DA,XINT,CNT,WH,XB,DB,C
     *,EPS,EPS1,ALEPS,TOT,PQ,D4,DD,PA
      DATA EPS,EPS1,ALEPS/1.D-6,1.D-78,-179.6016D0/
      Y=XF
      IF((XF.LE.1.0).AND.(XF.GE.0.0)) GO TO 5
      IER=129
      GO TO 9000
 5    IF((AF.GT.0.0).AND.(BF.GT.0.0)) GO TO 10
      IER=130
      GO TO 9000
 10   IER=0
      AA=AF
      BB=BF
      IF(XF.GT.0.5) GO TO 15
      INT=0
      GO TO 20
 15   INT=1
      TEMP=AA
      AA=BB
      BB=TEMP
      Y=1.D0-Y
```

```
20  IF(XF.NE.0.0.AND.XF.NE.1.0)GO TO 25
    PF=0.
    GO TO 60
25  IB=BB
    TEMP=IB
    PS=BB-FLOAT(IB)
    IF(BB.EQ.TEMP)PS=1.D0
    DA=AA
    DB=BB
    PX=DA*DLOG(Y)
    DD=DA+DB
    PQ=GAMMLN(DD)
    P1=GAMMLN(DA)
    C=GAMMLN(DB)
    D4=DLOG(DA)
    PA=PS+DA
    XB=PX+GAMMLN(PA)-GAMMLN(PS)-D4-P1
    IB=XB/ALEPS
    XINT=0.D0
    IF(IB.NE.0)GO TO 35
    XINT=DEXP(XB)
    CNT=XINT*DA
    WH=0.0D0
30  WH=WH+1.D0
    CNT=CNT*(WH-PS)*Y/WH
    XB=CNT/(DA+WH)
    XINT=XINT+XB
    IF(XB/EPS.GT.XINT) GO TO 30
35  TOT=0.D0
    IF(DB.LE.1.D0)GO TO 55
    XB=PX+DB*DLOG(1.D0-Y)+PQ-P1-DLOG(DB)-C
    IB=XB/ALEPS
    IF(IB.LE.0)IB=0
    C=1.D0/(1.D0-Y)
    CNT=DEXP(XB-DFLOAT(IB)*ALEPS)
```

```
      PS=DB
      WH=DB
   40 WH=WH-1.D0
      IF(WH.LE.0.0D0)GO TO 55
      PX=(PS*C)/(DA+WH)
      IF(PX.GT.1.D0)GO TO 45
      IF(CNT/EPS.LE.TOT.OR.CNT.LE.EPS1/PX)GO TO 55
   45 CNT=CNT*PX
      IF(CNT.LE.1.00) GO TO 50
      IB=IB-1
      CNT=CNT*EPS1
   50 PS=WH
      IF(IB.EQ.0)TOT=TOT+CNT
      GO TO 40
   55 PF=TOT+XINT
   60 IF(INT.NE.0)PF=1.-PF
      GO TO 9005
 9000 CONTINUE
 9005 RETURN
      END

      FUNCTION GAMMLN(XX)
      DOUBLE PRECISION SER,STP,TMP,X,Y,COF(6),XX
      DATA COF,STP/76.18009172947146D0,-86.50532032941677D0
     *,24.01409824083091D0,-1.231739572450155D0,.1208650973866179D-2
     *,-.5395239384953D-5,2.5066282746310005D0/
      X=XX
      Y=X
      TMP=X+5.5D0
      TMP=(X+0.5D0)*LOG(TMP)-TMP
      SER=1.000000000190015D0
      DO 10 J=1,6
      Y=Y+1.D0
      SER=SER+COF(J)/Y
   10 CONTINUE
```

```
      GAMMLN=TMP+LOG(STP*SER/X)
      RETURN
      END
```

2. กรณีข้อมูลมีค่าไม่ซ้ำกัน

```
************************************************************************
***** NOT REPEAT DATA — CLASSICAL BOOTSTRAP *****
************************************************************************
      INTEGER KE
      REAL Y(100),E(100),X(100,10),B(10),BT(10)
     *,RN(100),RN1(100),RN2(100)
     *,P(100),NORMAL,FNX(100),XMEAN(20),XSIGMA(20)
      REAL D01,D05,D10,W01,W05,W10
     *    ,SD01,SD05,SD10,SW01,SW05,SW10
      COMMON/SEED/IX,KN
     *    /SAMPLE/X,E
     *    /PROB/P
     *    /BETA/BT
      OPEN(3,FILE='A:OUTT13.DAT',STATUS='UNKNOWN')
      OPEN(4,FILE='A:INT13.DAT',STATUS='OLD')
      IX=25579
      NOFF=0
7     READ(4,*) MODEL,N,KE
      WRITE(3,*) MODEL,N,KE
      IF((MODEL.EQ.1).OR.(MODEL.EQ.2))THEN
        M=MODEL+1
        READ(4,*) XMEAN(2),XSIGMA(2)
        WRITE(3,*) XMEAN(2),XSIGMA(2)
      ELSE IF((MODEL.EQ.3).OR.(MODEL.EQ.4))THEN
        M=MODEL
        READ(4,*) XMEAN(2),XMEAN(3)
        WRITE(3,*) XMEAN(2),XMEAN(3)
        READ(4,*) XSIGMA(2),XSIGMA(3)
        WRITE(3,*) XSIGMA(2),XSIGMA(3)
      END IF
```

```
      WRITE(3,*)'REAL BETA'
      READ(4,*) (BT(I),I=1,M)
      KN=0
      NI=0
      SD01=0.0
      SD05=0.0
      SD10=0.0
      SW01=0.0
      SW05=0.0
      SW10=0.0
      D01=0.0
      D05=0.0
      D10=0.0
      W01=0.0
      W05=0.0
      W10=0.0
5     DO 1 I=1,N
       X(I,1)=1.0
       DO 3 II=2,M
       DMEAN=XMEAN(II)
       SIGMA=XSIGMA(II)
       X(I,II)=NORMAL(DMEAN,SIGMA)
       IF((II+1.EQ.M).AND.(MODEL.EQ.2))THEN
        X(I,3)=X(I,2)*X(I,2)
        GO TO 1
       ELSE IF((II+1.EQ.M).AND.(MODEL.EQ.4))THEN
        X(I,4)=X(I,2)*X(I,3)
        GO TO 1
       END IF
3      CONTINUE
1     CONTINUE
      DO 10 I=1,N
      Y(I)=0.0
      A=0.0
      DO 12 J=1,M
```

```
    A=A+X(I,J)*BT(J)
12  CONTINUE
   IF (KE.EQ.1)THEN
    EMEAN=0.0
    ESIGMA=1.0
    E(I)=NORMAL(EMEAN,ESIGMA)
    Y(I)=A+E(I)
   ELSE IF (KE.EQ.2)THEN
    EMEAN=0.0
    ESIGMA=SQRT(2.0)
    E(I)=NORMAL(EMEAN,ESIGMA)
    Y(I)=A+E(I)
   ELSE IF (KE.EQ.3)THEN
    EMEAN=0.0
    ESIGMA=SQRT(3.0)
    E(I)=NORMAL(EMEAN,ESIGMA)
    Y(I)=A+E(I)
   ELSE IF (KE.EQ.4)THEN
    EMEAN=0.0
    ESIGMA=0.5
    A1=NORMAL(EMEAN,ESIGMA)
    E(I)=EXP(A1)
    Y(I)=A+E(I)
   ELSE IF (KE.EQ.5)THEN
    EMEAN=0.0
    ESIGMA=1.0
    A1=NORMAL(EMEAN,ESIGMA)
    E(I)=EXP(A1)
    Y(I)=A+E(I)
   ELSE IF (KE.EQ.6)THEN
    EMEAN=0.0
    ESIGMA=1.5
    A1=NORMAL(EMEAN,ESIGMA)
    E(I)=EXP(A1)
    Y(I)=A+E(I)
```

```
      ELSE IF (KE.EQ.7) THEN
       ALP=2.0
       BTA=1.0
       E(I)=GAMMA1(ALP,BTA)
       Y=A+E(I)
      ELSE IF (KE.EQ.8) THEN
       ALP=3.0
       BTA=1.0
       E(I)=GAMMA1(ALP,BTA)
       Y=A+E(I)
      ELSE IF (KE.EQ.9) THEN
       ALP=4.0
       BTA=1.0
       E(I)=GAMMA1(ALP,BTA)
       Y=A+E(I)
      END IF
      P(I)=FLOAT(I)/FLOAT(N)
 10   CONTINUE
      CALL OLS(X,Y,B,M,N,SS1,YY,SUMY)
      L=N
*******************************************************************************
*                       Kolmogorov-Smirnov Test                               *
*******************************************************************************
      CALL RNX(L,M,N,X,Y,B,RN,RN1,RN2)
      CALL MAXDT(SUP,RN1,L)
      DCAL=SUP
      DO 68 I=1,L
      FNX(I)=0.0
      CN=0.0
      DO 67 J=1,L
      IF (X(J,2).LE.X(I,2)) THEN
      CN=CN+1.0
      END IF
 67   CONTINUE
      FNX(I)=CN/FLOAT(N)
```

```
 68  CONTINUE
     CALL BOOTS(FNX,L,M,N,D01,D05,D10,W01,W05,W10)
      IF (DCAL.GE.D01) THEN
        SD01=SD01+1.0
     END IF
     IF (DCAL.GE.D05) THEN
        SD05=SD05+1.0
     END IF
     IF (DCAL.GE.D10) THEN
        SD10=SD10+1.0
     END IF
*****************************************************************************
*                          Cramer-von Mises Test                            *
*****************************************************************************
     WCAL=0.0
     DO 70 I=1,L
     WCAL=WCAL+(RN2(I)*FNX(I))
 70  CONTINUE
     IF (WCAL.GE.W01) THEN
        SW01=SW01+1.0
     END IF
     IF (WCAL.GE.W05) THEN
        SW05=SW05+1.0
     END IF
     IF (WCAL.GE.W10) THEN
        SW10=SW10+1.0
     END IF
*****************************************************************************
*                   Type I Error  and Power of the Test                     *
*****************************************************************************
     NI=NI+1
     IF (NI.LT.1000) GO TO 5
     PD01=SD01/1000.
     PD05=SD05/1000.
     PD10=SD10/1000.
```

```
 PW01=SW01/1000.
 PW05=SW05/1000.
 PW10=SW10/1000.
 WRITE(3,*) '      KS-TEST        CvM-TEST'
 WRITE(3,*) '99%  ',PD01,PW01
 WRITE(3,*) '95%  ',PD05,PW05
 WRITE(3,*) '90%  ',PD10,PW10
*          Test Significant for Type I Error                    *
 IF((MODEL.EQ.1).OR.(MODEL.EQ.3))THEN
 Z01=0.01+1.645*SQRT(0.0099/FLOAT(NI))
 Z05=0.05+1.645*SQRT(0.0475/FLOAT(NI))
 Z10=0.10+1.645*SQRT(0.09/FLOAT(NI))
 IF((PD01.GE.0.0).AND.(PD01.LE.Z01))THEN
  WRITE(3,*)'KS-test Controlable type I Error at 0.01'
 END IF
 IF((PD05.GE.0.0).AND.(PD05.LE.Z05))THEN
  WRITE(3,*)'KS-test Controlable type I Error at 0.05'
 END IF
 IF((PD10.GE.0.0).AND.(PD10.LE.Z10))THEN
  WRITE(3,*)'KS-test Controlable type I Error at 0.10'
 END IF
 IF((PW01.GE.0.0).AND.(PW01.LE.Z01))THEN
  WRITE(3,*)'CVM-test Controlable type I Error at 0.01'
 END IF
 IF((PW05.GE.0.0).AND.(PW05.LE.Z05))THEN
  WRITE(3,*)'CVM-test Controlable type I Error at 0.05'
 END IF
 IF((PW10.GE.0.0).AND.(PW10.LE.Z10))THEN
  WRITE(3,*)'CVM-test Controlable type I Error at 0.10'
 END IF
 END IF
 NOFF=NOFF+1
 IF(NOFF.LT.63) GO TO 7
 CLOSE(3)
 CLOSE(4)
```

```
      STOP
      END
*********************************************************************************
*                      Subroutine for Estimated  Rn(x)                          *
*********************************************************************************
      SUBROUTINE RNX(L,M,N,X,Y,B,RN,RN1,RN2)
      DIMENSION X(100,10),Y(100),YH(100),B(10)
     *        ,R1(100),RN2(100),RN1(100),RN(100)
      DO 200 K=1,L
      RN(K)=0.0
      RN1(K)=0.0
      RN2(K)=0.0
      R1(K)=0.0
      DO 210 I=1,N
      YH(I)=0.0
      DO 220 J=1,M
      YH(I)=YH(I)+X(I,J)*B(J)
  220 CONTINUE
      IF (X(I,2).LE.X(K,2)) THEN
       R1(K)=R1(K)+(Y(I)-YH(I))
      ELSE
      END IF
  210 CONTINUE
      A=FLOAT(N)
      C=1.0/SQRT(A)
      RN(K)=C*R1(K)
      RN1(K)=ABS(RN(K))
      RN2(K)=RN(K)*RN(K)
  200 CONTINUE
      RETURN
      END
*********************************************************************************
*                      Subroutine for OLS Estimater                             *
*********************************************************************************
      SUBROUTINE OLS(X,Y,B,M,N,SS1,YY,SUMY)
```

```
      DIMENSION X(100,10),Y(100),B(10),XY(10),XX(10,10)
     *         ,XXI(10,10),YE(100)
      DO 20 I=1,M
      DO 22 J=1,M
        XY(J)=0.0
        XX(I,J)=0.0
        YY=0.0
        SUMY=0.0
      DO 25 K=1,N
        XX(I,J)=XX(I,J)+X(K,I)*X(K,J)
        XY(J)=XY(J)+X(K,J)*Y(K)
        YY=YY+Y(K)*Y(K)
        SUMY=SUMY+Y(K)
25    CONTINUE
22    CONTINUE
20    CONTINUE
      CALL INV(XX,XXI,M)
      DO 30 I=1,M
        B(I)=0.0
      DO 31 J=1,M
        B(I)=B(I)+XXI(I,J)*XY(J)
31    CONTINUE
30    CONTINUE
      DO 35 I=1,N
        YE(I)=0.0
        SS1=0.0
        DO 40 J=1,M
          YE(I)=YE(I)+X(I,J)*B(J)
          SS1=SS1+B(J)*XY(J)
40    CONTINUE
35    CONTINUE
      RETURN
      END
**********************************************************************************
*                   Subroutine for Ranking Data                                 *
```

```
********************************************************************************
    SUBROUTINE RANK(R)
    DIMENSION R(500)
290 IC=0
    DO 295 I=2,500
    II=I-1
    IF (R(II).LE.R(I)) GO TO 295
    Z=R(II)
    R(II)=R(I)
    R(I)=Z
    IC=1
295 CONTINUE
    IF (IC.EQ.1) GO TO 290
    RETURN
    END
********************************************************************************
*                 Subroutine for Sampling with Replacement                      *
********************************************************************************
    SUBROUTINE SWR(N,EBS)
    DIMENSION P(100),E(100),EBS(100),X(100,10)
    COMMON/SAMPLE/X,E/PROB/P/SEED/IX,KN
    DO 300 J=1,N
    CRN=RAND(IX)
    DO 305 I=1,N
    II=I-1
    IF (II.EQ.0) THEN
     X1=0.0
    ELSE
     X1=P(II)
    END IF
    X2=P(I)
    IF((CRN.GT.X1).AND.(CRN.LE.X2)) THEN
     EBS(J)=E(I)
     GOTO 300
    END IF
```

```
 305  CONTINUE
 300  CONTINUE
      RETURN
      END
*******************************************************************************
*       FUNCTION for generated random number Uniform(0,1)                    *
*******************************************************************************
      FUNCTION RAND(IX)
      IX = IX*16807
      IF (IX.LT.0) IX=(IX+2147483647)+1
      RAND=IX
      RAND=RAND*0.4656613E-9
      RETURN
      END
*******************************************************************************
*                        Subroutine for Inverse  Matrix                       *
*******************************************************************************
      SUBROUTINE INV(A,XXI,M)
      DIMENSION A(10,10),XXI(10,10)
      N=2*M
      N1=M+1
      M1=M-1
      DO 320 I=1,M
        M1=M1+1
        DO 320 J=N1,N
          M2=J-M1
          IF (M2.EQ.1) A(I,J)=1.0
          IF (M2.NE.1) A(I,J)=0.0
 320  CONTINUE
      DO 325 I=1,M
        DO 330 K=I,M
          IF(A(K,I).EQ.0.0) GO TO 330
          I1=K
          GO TO 340
 330  CONTINUE
```

```
340  IF(I1.EQ.I) GO TO 345
     DO 347 J=1,N
       E=A(I1,J)
       F=A(I,J)
       A(I,J)=E
       A(I1,J)=F
347  CONTINUE
345  D=A(I,I)
     DO 350 J=I,N
       A(I,J)=A(I,J)/D
350  CONTINUE
     DO 355 K=1,M
       IF(K.EQ.I) GO TO 355
       IF(A(K,I).EQ.0.0) GO TO 355
       C=A(K,I)
     DO 360 J=1,N
       A(K,J)=A(K,J)-(C*A(I,J))
360  CONTINUE
355  CONTINUE
325  CONTINUE
     DO 365 I=1,M
     DO 370 J=1,N
       K=J+M
       XXI(I,J)=A(I,K)
370  CONTINUE
365  CONTINUE
     RETURN
     END
**********************************************************************************
*                    Subroutine for Maximized Data                              *
**********************************************************************************
     SUBROUTINE MAXDT(SUP,R,L)
     DIMENSION R(100)
     SUP=R(1)
     DO 372 I=2,L
```

```
      IF (SUP.GE.R(I)) GO TO 372
      SUP=R(I)
 372  CONTINUE
      RETURN
      END

*********************************************************************************
*                Subroutine for Bootstrap Approximation                         *
*********************************************************************************
      SUBROUTINE BOOTS(FNX,L,M,N,D01,D05,D10
     *         ,W01,W05,W10)
      DIMENSION EBS(100),YBS(100),BB(10),X(100,10),FNX(100)
     *     ,RN1(100),RN2(100),DB(500),WB(500),BT(10)
     *     ,E(100),P(100),RN(100)
      REAL SUMY,YY,SUP,D01,D05,D10,W01,W05,W10
      COMMON/SAMPLE/X,E/PROB/P/BETA/BT
      D01=0.0
      D05=0.0
      D10=0.0
      W01=0.0
      W05=0.0
      W10=0.0
      DO 400 IB=1,500
      DB(IB)=0.0
      WB(IB)=0.0
      CALL SWR(N,EBS)
      DO 375 I=1,N
      YBS(I)=0.0
      DO 377 J=1,M
      YBS(I)=YBS(I)+(X(I,J)*BT(J))
 377  CONTINUE
      YBS(I)=YBS(I)+EBS(I)
 375  CONTINUE
      CALL OLS(X,YBS,BB,M,N,SS1,YY,SUMY)
      CALL RNX(L,M,N,X,YBS,BB,RN,RN1,RN2)
```

```
      CALL MAXDT(SUP,RN1,L)
      DB(IB)=SUP
      DO 380 I=1,L
      WB(IB)=WB(IB)+(RN2(I)*FNX(I))
380  CONTINUE
400  CONTINUE
      CALL RANK(DB)
      D01=DB(495)
      D05=DB(475)
      D10=DB(450)
      CALL RANK(WB)
      W01=WB(495)
      W05=WB(475)
      W10=WB(450)
      RETURN
      END
*********************************************************************************
*       FUNCTION for generated data from Normal Distribution              *
*********************************************************************************
      FUNCTION NORMAL(DMEAN,SIGMA)
      REAL NORMAL,RU1,RU2,U1,U2,PI
      COMMON/SEED/IX,KN
      PI=3.142857143
      IF(KN.EQ.1)GO TO 405
      RU1=RAND(IX)
      RU2=RAND(IX)
      U1=SQRT(-2*ALOG(RU1))*COS(2*PI*RU2)
      U2=SQRT(-2*ALOG(RU1))*SIN(2*PI*RU2)
      NORMAL=DMEAN+SIGMA*U1
      KN=1
      RETURN
405  NORMAL=DMEAN+SIGMA*U2
      KN=0
      RETURN
      END
```

```
******************************************************************************
*   FUNCTION for generated data from Lognormal Distribution                  *
******************************************************************************
    FUNCTION LOGNOR(DMEAN,SIGMA)
    REAL LOGNOR,NORMAL,E
    E=NORMAL(DMEAN,SIGMA)
    LOGNOR=EXP(E)
    RETURN
    END
```

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวทิพย์วัลย์ กันทอง เกิดเมื่อวันที่ 10 ธันวาคม พ.ศ. 2519 ที่จังหวัดพิษณุโลก สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต(เกียรตินิยมอันดับ 1) สาขาสถิติ จากมหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก ในปีการศึกษา 2540 และเข้าศึกษาต่อในหลักสูตรสถิติศาสตรมหาบัณฑิต ที่คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2541