รายการอ้างอิง

ภาษาไทย

ทัศนีย์ ชังเทศ, สมภพ ถาวรยิ่ง. การวิเคราะห์การถดถอยและสหสัมพันธ์. โรงพิมพ์มหาวิทยาลัยธรรมศาสตร์, 2530.

อัมพร ชาตบุษยมาส, การวิเคราะห์ความถดถอยเมื่อตัวแปรตามบางค่ามีค่าขาดหาย. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2530.

ภาษาอังกฤษ

หนังสือ

Dick London, FSA. Survival Models and Their Estimation. United States of America : ACTEX Publication,1988.

Elisa T. Lee. Statistical Methods for Survival Data Analysis. California : Wadsworth,1980.

J.F. Lawless. Statistical Models and Methods for Lifetime Data. New York : John Wiley & Sons,1982.

Robert V. Hogg & Stuart A. Klugman. Loss Distributions. New York : John Wiley & Sons,1984.

Roderick J.A. Little & Donald B. Rubin. Statistical Analysis with Missing Data. New York : John Wiley & Sons,1987.

Rupert G. Miller, Jr. Survival Analysis. New York : John Wiley & Sons, 1981.

Wayne Nelson. Applied Life Data Analysis. New York : John Wiley & Sons, 1982.

<u>บทความ</u>

Glenn Heller, "A comparison of estimators for regression with a censored response variable." Biometrika (1990): 77,3,515-520.

Helmut Schneider, "Estimation in linear models with censored data." Biometrika (1986) : 73,3,741-745.

I.R. James and P.J. Smith, "Consistency results for linear regression with censored data." The Annals of Statistics (1984) : 12,2, 590-600.

Jonathan Buckley, "Linear regression with cencored data." Biometrika (1979) : 66,3,429-436.

Josef Schmee and Gerald J. Hahn, "A simple method for regression analysis with censored data." Technometrics (1979) : 21,4, 417-432.

Murray Aitkin, "A note on the regression analysis of censored data." Technometrics (1981) : 23,2,161-163.

Rupert G. Miller, "Least squares regression with censored data." Biometrika (1976) : 63,3,449-464.

Rupert G. Miller and Jerry Halpern, "Regression with censored data." Biometrika (1982) : 69,3,521-531.

ภาคผนวก

ภาคผนวก ก

การสร้างตัวเลขสุ่ม (Random Number)

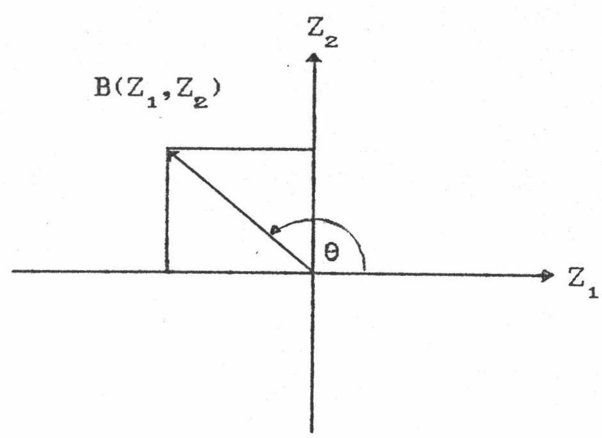
การสร้างลักษณะการแจกแจงแบบต่าง ๆ นั้น จะต้องอาศัยตัวเลขสุ่มเป็นพื้นฐานในการสร้าง สำหรับการวิจัยครั้งนี้จะใช้วิธีสร้างตัวเลขสุ่มตามวิธีของ White และ Schmidt (1975) ซึ่งขั้นตอนในการสร้างจะแสดงรายละเอียดด้วยฟังก์ชันต่อไปนี้

```
FUNCTION RAND(IX)
IX = IX*16807
IF (IX.LT.0) IX = IX+2147483647+1
RAND = IX
RAND = RAND*0.465661E-9
RETURN
END
```

ค่า IX จะเป็นค่า SEED หรือค่าเริ่มต้น ซึ่งจะต้องเป็นจำนวนเต็มบวกที่เป็นเลขคี่ RAND จะเป็นค่าของตัวเลขสุ่มที่มีค่าอยู่ระหว่าง 0 ถึง 1

การสร้างการแจกแจงแบบปกติ $N(\mu,\sigma^2)$:

การผลิตเลขสุ่มที่มีการแจกแจงแบบปกติใช้วิธีของ Box และ Muller (1958) โดยผลิตเลขสุ่มที่มีการแจกแจงปกติมาตรฐาน : N(0,1) พร้อมกัน 2 ค่า และแต่ละค่าเป็นอิสระกัน โดยใช้ตัวผลิต (Generator) Z_1 และ Z_2 พิจารณาดังรูปต่อไปนี้



พิจารณาจากรูปจะได้

$$Z_1 = B\cos(\Theta) \qquad (1)$$

$$Z_2 = B\sin(\Theta) \qquad (2)$$

เนื่องจาก $B^2 = Z_1^{\ 2} + Z_2^{\ 2}$ มีการแจกแจงไคสแควร์ด้วยระดับความเป็นอิสระ 2 และเทียบเท่าการแจกแจงเอกโปเนนเชียล ค่าเฉลี่ยเท่ากับ 2 โดยใช้วิธีการแปลงผกผัน (Inverse Transformation) สามารถสร้างเลขสุ่มที่มีการแจกแจงแบบเอกโปเนนเชียลได้ดังนี้

$$B = (-2\ \ln R)^{1/2} \qquad (3)$$

เมื่อ R เป็นเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง (0,1)

จากการสมมาตรของการแจกแจงปกติ (Normal Distribution) จะได้ว่ามุม Θ มีการแจกแจงแบบสม่ำเสมอระหว่าง 0 ถึง 2π เรเดียน และรัศมี B กับมุม Θ เป็นอิสระกัน จาก (1),(2),(3) เราสามารถสร้างเลขสุ่มที่มีการแจกแจงปกติมาตรฐาน จากเลขสุ่ม 2 ชุด R_1 และ R_2 กล่าวคือ

$$Z_1 = (-2\ \ln R_1)^{1/2}\cos(2\pi R_2)$$

$$Z_2 = (-2\ \ln R_1)^{1/2}\sin(2\pi R_2)$$

ซึ่ง R_1 และ R_2 เป็นตัวเลขสุ่มที่สร้างจากฟังก์ชัน FUNCTION RAND(IX) เมื่อได้ตัวเลขสุ่มที่มีการแจกแจงแบบปกติมาตรฐานแล้ว จะทำการแปลงค่าเลขสุ่มดังกล่าวโดยอาศัยฟังก์ชัน

$$EX_1 = \mu + \sigma Z_1$$

$$EX_2 = \mu + \sigma Z_2$$

ซึ่งจะได้ว่า EX_1 และ EX_2 มีการแจกแจงแบบปกติด้วยค่าเฉลี่ยเท่ากับ μ ค่าความแปรปรวนเท่ากับ σ^2 ($EX_i \sim N(\mu,\sigma^2)$; $i = 1,2$)

โปรแกรมย่อยที่ใช้ในการสร้างเลขสุ่มให้มีการแจกแจงปกติที่มีค่าเฉลี่ยเท่ากับ μ และมีค่าความแปรปรวนเท่ากับ σ^2 คือ SUBROUTINE NORMAL(RMEAN,VAR,EX) รายละเอียดโปรแกรมย่อยแสดงดังนี้

```
      SUBROUTINE NORMAL(RMEAN,VAR,EX)
      COMMON/SEED/IX,KK
      SD = SQRT(VAR)
      PI = 3.1415926
      IF (KK.EQ.1) GOTO 10
      RONE = RAND(IX)
      RTWO = RAND(IX)
      ZONE = SQRT(-2*ALOG(RONE))*COS(2*PI*RTOW)
      ZTWO = SQRT(-2*ALOG(RONE))*SIN(2*PI*RTOW)
      EX = ZONE*SD+RMEAN
      KK = 1
      GOTO 15
   10 EX = ZTWO*SD+RMEAN
      KK = 0
   15 RETURN
      END
```

<u>การสร้างการแจกแจงแบบสม่ำเสมอ</u> **U(a,b) :**

ฟังก์ชันความหนาแน่น และฟังก์ชันการแจกแจงสะสม แสดงได้ดังนี้

$$f(c) = \frac{1}{b-a} \quad , \quad a < c < b$$

$$F(c) = \begin{cases} (c-a)/(b-a) \;, & a < c < b \\ 1 \;, & b < c \end{cases}$$

โดย a   เป็นพารามิเตอร์กำหนดตำแหน่ง (Location Parameter)

b-a เป็นพารามิเตอร์กำหนดขนาด (Scale Parameter)

มีค่าคาดหวัง และค่าความแปรปรวนของการแจกแจงคือ

$$E(c) = (a+b) / 2$$

$$Var(c) = (b-a)^2 / 12$$

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบสม่ำเสมอ อาศัยการแปลงผกผันโดยมีขั้นตอนการสร้างดังนี้

ขั้นที่ 1 $F(c) = YFL$ โดยที่ YFL คือ ตัวเลขสุ่มแบบสม่ำเสมอช่วง (0,1)

ขั้นที่ 2 หาค่า $c = \{ (b-a)*YFL \} + a$

โปรแกรมย่อยที่ใช้ในการสร้างตัวเลขสุ่มให้มีการแจกแจงแบบสม่ำเสมอในช่วง (a,b) คือ SUBROUTINE UNIFOR(A,B,EX) ซึ่งรายละเอียดของโปรแกรมย่อยแสดงดังนี้

```
SUBROUTINE UNIFOR(A,B,EX)
COMMON/SEED/IX,KK
YFL = RAND(IX)
EX  = ( (B-A)*YFL ) + A
RETURN
END
```

<u>การสร้างการแจกแจงแบบปกติตัดปลายทางซ้าย</u> **(Left-Truncated Normal Distribution) TN(CO,μ,σ^2) :**

ถ้า c_o เป็นจุดขอบเขตที่ตัดปลายทางซ้าย มีฟังก์ชันความหนาแน่น และฟังก์ชันการแจกแจงสะสมแบบตัดปลายทางซ้าย ดังต่อไปนี้

$$f^*(c) = \frac{f(c)}{1 - F(c_o)} \quad ; \quad c > c_o$$

$$F^*(c) = \frac{F(c) - F(c_o)}{1 - F(c_o)}$$

โดยที่ $f(c) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}(c-\mu)^2\right)$

$$F(c) = \Phi\left[\frac{(c-\mu)}{\sigma}\right]$$

μ เป็นพารามิเตอร์กำหนดตำแหน่ง (Location Parameter),

$-\infty < \mu < \infty$

σ เป็นพารามิเตอร์กำหนดขนาด (Scale Parameter) ,$\sigma > 0$

มีค่าเฉลี่ย และค่าความแปรปรวนของการแจกแจงเป็น μ และ σ^2 ตามลำดับ

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบปกติตัดปลายทางซ้าย อาศัยการแปลงผกผัน โดยมีขั้นตอนการสร้างดังนี้

ขั้นที่ 1 หาค่า Z0 ณ จุดตัด c_o โดย $Z0 = (c_o - \mu) / \sigma$

ขั้นที่ 2 ค่า Z0 จากขั้นที่ 1 นำมาหาการแจกแจงสะสมของ c_o คือ $F(c_o)$ โดยเรียกใช้โปรแกรมย่อย SUBROUTINE PP(Z0,P0)

ขั้นที่ 3 $F^*(c)$ = YFL โดยที่ YFL คือ ตัวเลขสุ่มแบบสม่ำเสมอช่วง (0,1)

ขั้นที่ 4 หาค่า $F(c) = F(c_o) + (1 - F(c_o))$*YFL = PHI

ขั้นที่ 5 กำหนด $F(c) = \Phi[(c-\mu)/\sigma]$ = PHI

ขั้นที่ 6 จากขั้นที่ 5 นำค่า PHI มาหาส่วนกลับของการแจกแจงสะสมของการแจกแจงแบบปกติ (Inverse of the CDF. of the Normal Distribution) เพื่อหาค่า ZZ โดยใช้ฟังก์ชัน FUNCTION VNORM(PHI,IFAULT)

ขั้นที่ 7 หาค่า c ได้ดังนี้เมื่อ $(c-\mu)/\sigma$ = ZZ

ดังนั้น $c = (ZZ*\sigma) + \mu$

โปรแกรมย่อย ที่ใช้ในการสร้างตัวแปรสุ่ม ให้มีการแจกแจงแบบปกติตัดปลายทางซ้าย คือ SUBROUTINE TNORM(C0,RM,SD,EX) ซึ่งรายละเอียดของโปรแกรมย่อยและฟังก์ชันแสดงได้ดังนี้

```
      SUBROUTINE TNORM(CO,RM,SD,EX)
      DOUBLE PRECISION PHI,IFAULT
      COMMON/SEED/IX,KK
      ZO = (CO-RM)/SD
      YFL = RAND(IX)
      CALL PP(ZO,PO)
      PROB = PO + YFL*(1-PO)
      IFAULT = 0.
      ZZ = VNORM(PROB,IFAULT)
      EX = (ZZ*SD) + RM
      RETURN
      END

      FUNCTION VNORM(PHI,IFAULT)
      DOUBLE PRECISION PHI
     *   ,IFAULT,PLIM,PO,P1,P2,P3,P4,Q0,Q1,Q2,Q3,Q4,P,VTEMP
C VNORM RETURNS THE INVERSE OF THE CDF OF THE NORMAL DISTRIBUTION
C IF USES A RATIONAL APPROXIMATION WHICH SEEMS TO HAVE A REATIVE
C ACCURACY OF ABOUT 5 DECIMAL PLACES.
C REF.: DENNEDY AND GENTLE, STATISTICAL COMPUTING,DEKKER,1980.
C INPUT:
C PHI = PROBABILITY,0 <= PHI <= 1.
C OUTPUTS:
C F INVERSE OF PHI, I.E., A VALUE SUCH THAT
C PROB(X <= VNORM) = PHI.
C IFAULT = 6 IF PHI OUT OF RANGE,ELSE 0
```

```
      DATA PLIM /1.0D-18/
      DATA P0/-0.322232431088D0/,P1/-1.0/,P2/-0.342242088547D0/
      DATA P3 / -0.0204231210245D0/, P4/-0.453642210148D-4/
      DATA Q0/0.099348462606D0/, Q1/0.588581570495D0/
      DATA Q2/0.531103462366D0/,Q3/0.10353775285D0/
      DATA Q4/0.38560700634D-2/
      IFAULT = 0
      P = PHI
      IF (P .GT. 0.5) P = 1. -P
      IF (P .GE. PLIM) GOTO 100
C THIS IS AS FAR OUT IN THE TAILS AS WE GO
      VTEMP = 8.
C CHECK FOR INPUT ERROR
      IF (P .LT. 0.) GOTO 9000
      GOTO 200
  100 Y = DSORT(-DLOG(P*P))
      VTEMP = Y + ((((Y*P4 + P3)*Y+P2)*Y+P1)*Y+P0)/
     *    ((((Y*Q4+Q3)*Y+Q2)*Y+Q1)*Y+Q0)
  200 IF (PHI .LT. 0.5) VTEMP = - VTEMP
      VNORM = VTEMP
      RETURN
 9000 IFAULT = 6
      RETURN
      END
      SUBROUTINE PP(Z0,P0)
      R = EXP(-Z0*Z0/2.)/2.5066282746
      WW = 1./(1.+0.33267*ABS(Z0))
```

```
P = 1.-R*(0.4361836*WW-0.1201676*(WW**2)+0.937298*(WW**3))
IF (ZO .GE. 0) THEN
   PO = P
ELSE
   PO = 1.-P
ENDIF
RETURN
END
```

<u>การสร้างการแจกแจงแบบไวบูลล์ตัดปลายทางซ้าย</u> **(Left-Truncated Weibull Distribution) TW(CO,WAL,WBE,EX) :**

ถ้า c_o เป็นจุดขอบเขตที่ตัดปลายทางซ้าย มีฟังก์ชันความหนาแน่น และฟังก์ชันการแจกแจงสะสมแบบตัดปลายทางซ้าย ดังต่อไปนี้

$$f^*(c) = \frac{f(c)}{1 - F(c_o)} \quad ; \quad c > c_o$$

$$F^*(c) = \frac{F(c) - F(c_o)}{1 - F(c_o)}$$

โดยที่ $f(c) = \alpha\beta\, c^{\beta-1} \exp(-\alpha c^{\beta})$

$F(c) = 1 - \exp(-\alpha c^{\beta})$

β เป็นพารามิเตอร์กำหนดรูปร่าง (Shape Parameter), $\beta>0$

α เป็นพารามิเตอร์กำหนดขนาด (Scale Parameter), $\alpha>0$

มีค่าคาดหวัง และค่าความแปรปรวนของการแจกแจง คือ

$$E(c) = \alpha^{(-1/\beta)} * [\ \Gamma(\ 1 + 1/\beta\ )\ ]$$

$$Var(c) = \alpha^{(-2/\beta)} * [\ \Gamma(\ 1 + 2/\beta\ )\ ]$$

การสร้างตัวแปรสุ่มให้มีการแจกแจงแบบไวบูลล์ตัดปลายทางซ้าย อาศัยการแปลงผกผัน โดยมีขั้นตอนการสร้างดังนี้

ขั้นที่ 1 หาฟังก์ชันการแจกแจงสะสมของ c_o คือ $F(c_o)$

$$F(c_o) = 1 - \exp(-\alpha c_o^{\beta})$$

ขั้นที่ 2 $F^*(c)$ = YFL โดยที่ YFL คือ ตัวเลขสุ่มแบบสม่ำเสมอช่วง (0,1)

ขั้นที่ 3 หาค่า $F(c) = F(c_o) + (1 - F(c_o))*YFL$ = WPRO

ขั้นที่ 4 หาค่าของ c ในเทอมของ WPRO ได้ดังนี้

$$c = \{ [-\ln(1-WPRO)]^{1/\beta} \} / \alpha$$

โปรแกรมย่อย ที่ใช้ในการสร้างตัวแปรสุ่ม ให้มีการแจกแจงแบบไวบูลล์ตัดปลายทางซ้าย คือ SUBROUTINE TWEI(CO,WAL,WBE,EX) ซึ่งรายละเอียดของโปรแกรมย่อยแสดงได้ดังนี้

```
SUBROUTINE TWEI(CO,WAL,WBE,EX)
COMMOND/SEED/IX,KK
FCO = 1 - ( EXP(-WAL*(CO**WBE)) )
YFL = RAND(IX)
WPRO = FCO + ( (1-FCO)*YFL )
B1  = 1/WBE
B2 = -( ALOG(1-WPRO) )
EX = (B2/WAL)**B1
RETURN
END
```

การสร้างโปรแกรมย่อยเรียงลำดับข้อมูล

โปรแกรมย่อยที่ใช้ในการเรียงลำดับข้อมูล SUBROUTINE SORT(EM,IS,EM1)

```
      SUBROUTINE SORT(EM,IS,EM1)
      DIMENSION EM(400),EM1(400)
      DO 10 I = 1,IS
      EM1(I) = EM(I)
   10 CONTINUE
      K = IS-1
      DO 5 I = 1,K
      K1 = I+1
      DO 5 J = K1,IS
      IF (EM1(I).LE.EM1(J)) GOTO 5
      S = EM1(I)
      EM1(I) = EM1(J)
      EM1(J) = S
    5 CONTINUE
      RETURN
      END
```

<u>การสร้างโปรแกรมย่อยเพื่อหาฟังก์ชันแกมมา</u> ($\Gamma(n)$)

โปรแกรมย่อยที่ใช้ในการหาฟังก์ชันแกมมา SUBROUTINE FAC(N,GAM)

```
   SUBROUTINE FAC(N,GAM)
   IF ( (N .EQ. 0) .OR. (N .EQ. 1) ) THEN
      GAM = 1
      GOTO 20
   ENDIF
   GAM = 1
   DO 10 I = 2,N
   GAM = I*GAM
10 CONTINUE
20 RETURN
   END
```

ภาคผนวก ข

```
C*******************************************************************C
C***                      MAIN  PROGRAM                          ***C
C***    ESTIMATING PARAMETERS IN SIMPLE LINEAR REGRESSION WITH   ***C
C***               A CENSORED RESPONSE VARIABLE                  ***C
C*******************************************************************C
      DIMENSION YGEN(400),XGEN(400),Y1(400),X1(400),Y2(400),E(400)
     *          ,B(2),C(400),C1(400),NUM(400),XML(400),YML(400)
     *          ,XLS(400),YLS(400),XBJ(400),YBJ(400),BBJ1(1000)
     *          ,BBJ2(1000),WML(400),WLS(400),YB(400),BLS(2),BML(2)
     *          ,BBJ(2),DELTA(400),YHLS(400),YHML(400),YHBJ(400)
      COMMON/SEED/IX,KK
      DATA NM,RMEAN1,VAR1,RMEAN2,VAR2/
     *10,20.,60.,0.,16./
      ICEN = 40
      M = NM*ICEN/100
      N = NM - M
      NC1 = N + 1
      IX=65479
      KK=0
      DO 10 I=1,100
      A=RAND(IX)
   10 CONTINUE
      TLS   = 0
      TML   = 0
      TBJ   = 0
      SBLS1 = 0
      SBLS2 = 0
      SBML1 = 0
      SBML2 = 0
      SBBJ1 = 0
      SBBJ2 = 0
      BETA0 = 2
      BETA1 = 1
      TNUM = 0
```

```
      RA = (BETAO + RMEAN1) - (1.5*SQRT(VAR1)) - (1.5*SQRT(VAR2))
      RB = (BETAO + RMEAN1) + (1.5*SQRT(VAR1)) + (1.5*SQRT(VAR2))
      RMT = (RA+RB)/2
      INUM = 1000
      DO 300 INUM1=1,INUM
C*********************************************************************C
C***              GENERATE UNCENSORED AND CENSORED DATA            ***C
C*********************************************************************C
   20 KU = 0
      KC = 0
      KT = 0
   30 CALL GENT(BETAO,BETA1,RMEAN1,VAR1,RMEAN2,VAR2,TU1,XU10)
      IF ( (TU1 .LE. 0) .OR (XU10 .LE. 0) )THEN
         GOTO 30
      ELSE
         TU = TU1
         XU = XU10
      ENDIF
C*********************************************C
C***      (1.1) C(I) IS UNFORM DISTR.      ***C
C*********************************************C
      CALL UNIFOR(RA,RB,EX8)
         CU = EX8
C*********************************************C
C*** (1.2) C(I) IS TRUNCATED NORMAL DISTR. ***C
C*********************************************C
C     XXO = RA
C     CALL TNORM(XXO,RMT,(2*RMT),EX8)
C        CU = EX8
C*********************************************C
C*** (1.3) C(I) IS TRUNCATED WEIBULL DISTR.***C
C*********************************************C
C     XXO = RA
C     WBE = 2.0
C     IG2 = 1/WBE
C     CALL FAC(IG2,GAM2)
C     WAL = (GAM2/RMT)**WBE
```

```
C     CALL TWEI(XXO,WAL,WBE,EX8)
C         CU = EX8
C*********************************************C
C***      (1.4) C(I) IS LINEAR FUNCTION     ***C
C*********************************************C
C     RMEAN3 = 0
C     VAR3 = 25
C     CALL NORMAL(RMEAN1,VAR1,XC)
C     CALL NORMAL(RMEAN3,VAR3,EC)
C         CU = BETAO + ( BETA1*XC ) + EC
C*********************************************C
C***        Y(I) = MIN( T(I) ≤ C(I) )       ***C
C*********************************************C
      KT = KT+1
      IF (TU .LE. CU) THEN
         KU = KU+1
         YGEN(KT)  = TU
         XGEN(KT)  = XU
         DELTA(KT) = 1
      ELSE
         KC = KC+1
         YGEN(KT)  = CU
         XGEN(KT)  = XU
         DELTA(KT) = 0
      ENDIF
      IF (KC .EQ. M) THEN
         NU = KT+1
         DO 40 I = NU,NM
         CALL GENT(BETAO,BETA1,RMEAN1,VAR1,RMEAN2,BAR2,TU2,XU2)
         YGEN(I)  = TU2
         XGEN(I)  = XU2
         DELTA(I) = 1
   40 CONTINUE
         GOTO 70
      ELSE
         GOTO 50
      ENDIF
```

```
   50 IF (KU .EQ. N) THEN
         NC = KT+1
         DO 60 I = NC,NM
C************** CENSORING DISTR. *************C
         CALL UNIFOR(RA,RB,CU2)
C        CALL TNORM(XX0,RMT,(2*RMT),CU2)
C        CALL TWIE(XX0,WAL,WBE,CU2)
C********************************************C
         CALL NORMAL(RMEAN1,VAR1,XU2)
            YGEN(I)  = CU2
            XGEN(I)  = XU2
            DELTA(I) = 0
C************** LINEAR FUNCTION **************C
C        CALL NORMAL(RMEAN3,VAR3,EC)
C           YGEN(I)  = BETA0 + (BETA1*XU2) + EC
C           XGEN(I)  = XU2
C           DELTA(I) = 0
   60 CONTINUE
         GOTO 70
      ELSE
         GOTO 30
      ENDIF
   70 DO 80 I = 1,NM
      Y2(I) = YGEN(I)
   80 CONTINUE
      DO 90 I = 1,NM
      XLS(I) = XGEN(I)
      YLS(I) = YGEN(I)
      XML(I) = XGEN(I)
      YML(I) = YGEN(I)
      XBJ(I) = XGEN(I)
      YBJ(I) = YGEN(I)
   90 CONTINUE
      CALL BJ(XBJ,NM,N,M,YBJ,DELTA,INUM1,TNUM,YB,BBJ,YHBJ)
      IF ( BBJ(2) .EQ. 0 ) THEN
         GOTO 20
      ELSE
```

```
         BBJ1(INUM1) = BBJ(1)
         BBJ2(INUM1) = BBJ(2)
         CALL LS(XLS,NM,N,M,YLS,DELTA,WLS,BLS,YHLS)
         CALL ML(XML,NM,N,M,YML,DELTA,WML,BML,YHML)
      ENDIF
      SBLS1 = SBLS1 + BLS(1)
      SBLS2 = SBLS2 + BLS(2)
      SBML1 = SBML1 + BML(1)
      SBML2 = SBML2 + BML(2)
      SBBJ1 = SBBJ1 + BBJ1(INUM1)
      SBBJ2 = SBBJ2 + BBJ2(INUM1)
      SLS = 0
      SML = 0
      SBJ = 0
      DO 100 I=1,NM
      IF (DELTA(I) .EQ. 1) THEN
         SLS = SLS + ( Y2(I)-YHLS(I) )**2
         SML = SML + ( Y2(I)-YHML(I) )**2
      IF ( BBJ(2) .EQ. 0 ) THEN
         SBJ  = 0.
      ELSE
         SBJ  = SBJ + ( Y2(I)-YHBJ(I) )**2
      ENDIF
      ENDIF
  100 CONTINUE
      RLS = SQRT(SLS/N)
      RML = SQRT(SML/N)
      IF ( SBJ .EQ. 0 ) THEN
         RBJ = 0.
      ELSE
         RBJ = SQRT(SBJ/N)
      ENDIF
      TLS = TLS + RLS
      TML = TML + RML
      TBJ = TBJ + RBJ
  300 CONTINUE
      ITNUM  = INUM + TNUM
```

```
C****************** AVERAGE BETA *********************C
      SBLS1  = SBLS1/INUM
      SBLS2  = SBLS2/INUM
      SBML1  = SBML1/INUM
      SBML2  = SBML2/INUM
      SBBJ1  = SBBJ1/(INUM)
      SBBJ2  = SBBJ2/(INUM)
C************* AVERAGE RMSE OF ESTIMATE *************C
      RMLS   = TLS/INUM
      RMML   = TML/INUM
      RMBJ   = TBJ/INUM
C************* COMPUTE % NOT CONVERGENCE ************C
      TT = (TNUM/ITNUM)*100
      WRITE (6,110) ITNUM,INUM,TNUM,TT
  110 FORMAT(' CENSORED,TOTAL,#UNCEN,#CEN,%CEN ',2I5,2F5.0)
      WRITE (6,120)
  120 FORMAT('                OLS     ','     MLE     ', '    BUC  ')
      WRITE (6,130) SBLS1,SBML1,SBBJ1
  130 FORMAT(' BO        ',3F11.3)
      WRITE (6,140) SBLS2,SBML2,SBBJ2
  140 FORMAT(' B1        ',3F11.3)
      WRITE (6,150) RMLS,RMML,RMBJ
  150 FORMAT(' RMSE-FO. ',3F11.3)
      STOP
      END
C*****************************************************************C
C***                  STOP MAIN PROGRAM                        ***C
C*****************************************************************C
C***                 LEAST SQUARE METHOD                       ***C
C*****************************************************************C
      SUBROUTINE LS(,XX,IS,N,M,YY,DELTA,WL,BL,YHL)
      DIMENSION  XX(400),YY(400),WL(400),BL(2),DELTA(400),YHL(400)
      CALL OLS(XX,IS,YY,AMSE,BL)
      DO 10 I=1,IS
      IF (DELTA(I) .EQ. 1) THEN
         YHL(I)=BL(1)+(BL(2)*XX(I))
      ENDIF
```

```
   10 CONTINUE
      RETURN
      END
C*****************************************************************C
C***              MAXIMUM LIKELIHOOD ESTIMATION METHOD        ***C
C*****************************************************************C
      SUBROUTINE ML(XX,NM,N,M,YY,DELTA,WM,BM,YHM)
      DIMENSION YY(400) ,XX(400),B(2),E(400),SMLE(400),BOLS(2)
     *       ,SIGMA(400),BM(2),BO(400),B1(400),Z(400),ZBO(400)
     *       ,ZB1(400),FZ(400),FXZ(400),C(1,200),AM(400),WM(400)
     *       ,ZSIGMA(400),DELTA(400),YHM(400),HZ(400)
      K=1
      CALL OLS(XX,NM,YY,AMSE,BOLS)
      BO(K) = BOLS(1)
      B1(K) = BOLS(2)
      SIGMA(K) = (AMSE)
      DO 10 I=1,NM
      IF (DELTA(I) .EQ. 0)THEN
         C(1,I) =YY(I)
      ENDIF
   10 CONTINUE
   20 K=K+1
      KK=K-1
      SUM=0
      DO 30 I=1,NM
      IF (DELTA(I) .EQ. 1) THEN
         AM(I) = BO(KK) + (B1(KK) * XX(I))
         SUM = SUM + (YY(I) - AM(I))**2
      ENDIF
   30 CONTINUE
      SUM1 = 0
      DO 40 I = 1,NM
      IF (DELTA(I) .EQ. 0) THEN
      AM(I) = BO(KK) + (B1(KK) * XX(I))
         Z(I) = ( C(1,I)-AM(I) )/SIGMA(KK)
         IF (Z(I).EQ.0) THEN
```

```
      IF (Z(I).GE.3) THEN
         FZ(I) = 0
         FXZ(I) = 1
      ELSE
         FZ(I) = EXP(-Z(I)*Z(I)/2.)/  2.506628275
         ZI=Z(I)
         CALL PP(ZI,PXZ1)
         FXZ(I)=PXZ1
      ENDIF
   ELSE
      IF (Z(I).LE.-3) THEN
         FZ(I) = 0
         FXZ(I) = 0
      ELSE
         FZ(I) = EXP(-Z(I)*Z(I)/2.)/  2.506628275
         ZI=Z(I)
         CALL PP(ZI,PXZ1)
         FXZ(I)=PXZ1
      ENDIF
   ENDIF
IF ( (FZ(I) .EQ. 0) .OR. (1.-FXZ(I) .EQ. 0) ) THEN
   HZ(I) = 0
ELSE
   HZ(I) = FZ(I)/(1.-FXZ(I))
ENDIF
IF (HZ(I).EQ.0) THEN
   WM(I) = AM(I)
ELSE
   WM(I)= AM(I)+ (SIGMA(KK)*HZ(I))
ENDIF
YY(I) = WM(I)
IF (HZ(I).EQ.0) THEN
   SUM1 = SUM1
ELSE
   SUM1 = SUM1+ (1.+ Z(I)* HZ(I)
ENDIF
ENDIF
```

```
   40 CONTINUE
      SUM1 = SUM1 * SIGMA(KK)**2
      SMLE(K) = SQRT((SUM+SUM1)/NM)
      SIGMA(K) = (SMLE(K))
      CALL OLS(XX,NM,YY,AMSE,BOLS)
      BO(K) = BOLS(1)
      B1(K) = BOLS(2)
C*****************  TEST FOR CONVERGENCE  ******************C
      IF (K.GE.3) THEN
         II = K-2
         IF ( (BO(K).EQ.BO(II)).OR.(B1(K).EQ.B1(II)).OR.
     *   (SIGMA(K).EQ.SIGMA(II)) ) THEN
             BO(K) = ( BO(K) + BO(KK) ) / 2
             B1(K) = ( B1(K) + B1(KK) ) / 2
             SIGMA(K) = ( SIGMA(K) + SIGMA(KK) ) / 2
             GOTO 60
          ELSE
             GOTO 50
          ENDIF
      ENDIF
   50 BBB =BO(K)-BO(KK)
      ZBO(K) = ABS(BBB)
      ZB1(K) = ABS(B1(K)-B1(KK))
      ZSIGMA(K) = ABS(SIGMA(K)-SIGMA(KK))
      IF ( (ZBO(K).LE.0.001) .AND. (ZB1(K).LE.0.001) .AND.
     *(ZSIGMA(K).LE.0.001)    ) THEN
      GOTO 60
      ENDIF
      GOTO 20
   60 RMSE =SIGMA(K)
      BM(1)=BO(K)
      BM(2)=B1(K)
      DO 70 I = 1,NM
      IF (DELTA(I) .EQ. 1) THEN
         YHM = BM(1) + (BM(2) * XX(I))
      ENDIF
   70 CONTINUE
```

```
      RETURN
      END
C*****************************************************************C
C***                  BUCKLEY AND JAMES METHOD                 ***C
C*****************************************************************C
      SUBROUTINE BJ(XX,NM,N,M,YY,DELTA,INUM1,TNUM,YBA,BB,YHB)
      DIMENSION YY(400),XX(400),E(400),ESORT(400),FC(400),BB(2)
     *          ,B0(400),B1(400),NUM2(400),NR(400),SI(400),S(400)
     *          ,F(400),V(400),W(400),SCI(400),SC(400),NR1(400)
     *          ,AAA(800),BBB(800),YBA(400),DELTA(400),YHB(400)
     *          ,DEL(400),YY1(400),XX1(400)
      K = 1
      K2 = K + 1
      SUX = 0
      SUY = 0
      DO 10 I = 1,NM
      IF (DELTA(I) .EQ. 1) THEN
         SUX = SUX + XX(I)
         SUY = SUY + YY(I)
      ENDIF
   10 CONTINUE
      XBAR = SUX / N
      YBAR = SUY / N
      SUXY = 0
      SUXX = 0
      DO 20 I = 1,NM
      IF (DELTA(I) .EQ. 1) THEN
         SUXY = SUXY + YY(I) * ( XX(I) - XBAR )
         SUXX = SUXX + ( XX(I) - XBAR )**2
      ENDIF
   20 CONTINUE
      B1(K) = SUXY / SUXX
      B0(K) = YBAR - ( B1(K) * XBAR )
   30 DO 40 I = 1,NM
         E(I) = ABS( YY(I) - B1(K) * XX(I) )
   40 CONTINUE
      CALL SORT(E,NM,ESORT)
```

```
      DO 50 I = 1,NM
      DO 60 J = 1,NM
      IF ( ESORT(I) .EQ. E(J) ) THEN
         NUM2(I) = J
      ENDIF
   60 CONTINUE
   50 CONTINUE
      DO 70 I = 1,NM
      DEL(I) = DELTA(NUM2(I))
      YY1(I) = YY(NUM2(I))
      XX1(I) = XX(NUM2(I))
   70 CONTINUE
C****************** COMPUTE W(e_1,β̂) = V(I) ******************C
      DO 77 I = 1,NM
      IF ( DEL(I) .EQ. 1 ) THEN
         NR(I) = 0
      ELSE
         NR(I) = I
      ENDIF
   77 CONTINUE
      DO 80 I = 1,NM
      IF (NR(I) .EQ. 0 ) THEN
         AAA(I) = NM
         BBB(I) = NM + 1
         SI(I) = AAA(I) / BBB(I)
      ELSE
         AAA(I) = NM - NR(I)
         BBB(I) = NM - NR(I) + 1
         SI(I) = AAA(I) / BBB(I)
      ENDIF
   80 CONTINUE
      DO 90 I = 1,NM
      IF (I .EQ. 1) THEN
         S(I) = SI(I)
         F(I) = 1. - S(I)
         V(I) = F(I)
      ELSE
```

```
         II = I-1
         S(I) = S(I-1) * SI(I)
         F(I) = 1. - S(I)
         V(I) = F(I) - F(II)
      ENDIF
   90 CONTINUE
C****************** COMPUTE W*(e1,β̂) = V(I) *****************C
      IF (NR(NM) .EQ. 0) THEN
         SUW = 0
      DO 100 I = 1,NM
      IF (NR(I) .EQ. 0) THEN
         SUW = SUW
      ELSE
         SUW = SUW + V(I)
      ENDIF
  100 CONTINUE
      DO 110 I = 1,NM
      V(I) = V(I) / SUW
  110 CONTINUE
      ENDIF
C***************** COMPUTE F(e1,-β̂XI) = FC(I) ***************C
      DO 120 I = 1,NM
      IF (NR(I) .EQ. 0) THEN
         NR1(I) = I
      ELSE
         NR1(I) = 0
      ENDIF
  120 CONTINUE
      DO 130 I = 1,NM
      IF ( NR1(I) .EQ. 0 ) THEN
         AAA(I) = NM
         BBB(I) = NM + 1
         SCI(I) = AAA(I) / BBB(I)
      ELSE
         AAA(I) = NM - NR1(I)
         BBB(I) = NM - NR1(I) + 1
      SCI(I) = AAA(I) / BBB(I)
```

```
      ENDIF
  130 CONTINUE
      DO 140 I = 1,NM
      IF (I .EQ. 1) THEN
         SC(I) = SCI(I)
         FC(I) = 1. - SCI(I)
      ELSE
         II = I-1
         SC(I) = SC(I-1) * SCI(I)
         FC(I) = 1. - SC(I)
      ENDIF
  140 CONTINUE
C************************ COMPUTE WEITHTED ********************C
      DO 150 I = 1,NM
      IF ( FC(I) .EQ. 1 ) THEN
         W(I) = 0
      ELSE
         W(I) = V(I) / ( 1-FC(I) )
      ENDIF
  150 CONTINUE
C***************** ESTIMATING CENSORED DATA *****************C
      SUWB = 0
      DO 160 I = 1,NM
      IF (DEL(I) .EQ. 1) THEN
         SUWB = SUWB + W(I) * ( YY1(I) - B1(K) * XX1(I) )
      ENDIF
  160 CONTINUE
      DO 170 I = 1,NM
      IF (DELTA(I) .EQ. 0) THEN
         YBA(I) = ( B1(K) * XX(I) ) + SUWB
         YY(I) = YBA(I)
      ENDIF
  170 CONTINUE
      XBNEW = 0
      DO 180 I = 1,NM
      XBNEW = XBNEW + XX(I)
  180 CONTINUE
```

```
      XBNEW = XBNEW / NM
      SUXX = 0
      DO 190 I = 1,NM
      SUXX = SUXX + ( XX(I) - XBNEW ) **2
  190 CONTINUE
      SUYUC = 0
      SUYC  = 0
      YBNEW = 0
      DO 200 I = 1,NM
      IF (DELTA(I) .EQ. 1) THEN
         SUYUC = SUYUC + YY(I) * ( XX(I) - XBNEW )
         YBNEW = YBNEW + YY(I)
      ELSE
         SUYC = SUYC + YBA(I) * ( XX(I) - XBNEW )
         YBNEW = YBNEW + YBA(I)
      ENDIF
  200 CONTINUE
      K2 = K+1
      B1(K2) = ( SUYUC + SUYC ) / SUXX
C************************ TEST FOR CONVERGENCE *********************C
      IF ( (K2).LE. 20) THEN
         IF ( K2 .GE. 3 ) THEN
               II = K2-2
               LL = K2-1
               IF ( B1(K2) .EQ. B1(II) ) THEN
                    B1(K2)= ( B1(K2) + B1(LL) ) / 2
                    GOTO 230
               ELSE
                    GOTO 220
               ENDIF
           ENDIF
  220 ERROR = ABS( B1(K) - B1(K2) )
         IF ( ERROR .LE. 0.001) THEN
               GOTO 230
           ELSE
               K = K + 1
               GOTO 30
```

```
         ENDIF
      ELSE
         TNUM= TNUM + 1
         B1(K2) = 0
      ENDIF
  230 IF ( B1(K2) .EQ. 0 ) THEN
         BO(K2) = 0
      ELSE
         BO(K2) = ( YBNEW / NM ) - ( B1(K2) * XBNEW )
      ENDIF
      BB(1) = BO(K2)
      BB(2) = B1(K2)
      DO 240 I = 1,NM
      IF(DELTA(I) .EQ. 1) THEN
         YHB(I) = BB(1) + (BB(2) * XX(I))
      ENDIF
  240 CONTINUE
      RETURN
      END
C*********************************************************************C
C***                SUBROUTINE ODINARY LEAST SQUARE                ***C
C*********************************************************************C
      SUBROUTINE OLS(XX,IS,YY,AMSE1,BE)
      DIMENSION  BE(2),YY(400),XX(400),YNEW(400),XNEW(400)
      SUMX = 0
      SUMY = 0
      SUMYY = 0
      SUMXX = 0
      DO 10 I=1,IS
      SUMX = SUMX + XX(I)
      SUMY = SUMY + YY(I)
   10 CONTINUE
      XBAR = SUMX/IS
      YBAR = SUMY/IS
      DO 20 I =1 ,IS
      XNEW(I) = XX(I)-XBAR
      YNEW(I) = YY(I)-YBAR
```

```
   20  CONTINUE
       SUY = 0
       SUX = 0
       SUXY = 0
       DO 30 I=1,IS
       SUX = SUX + XNEW(I)**2
       SUY = SUY + YNEW(I)**2
       SUXY = SUXY + XNEW(I)*YNEW(I)
   30  CONTINUE
       BE(2) = SUXY/SUX
       BE(1) = YBAR - BE(2)*XBAR
       AMSE1 = SQRT( (SUY - BE(2)*SUXY) / (IS-2) )
       RETURN
       END
C*********************************************************************C
C***                     SUBROUTINE NORMAL                         ***C
C*********************************************************************C
       SUBROUTINE NORMAL(RMEAN,VAR,EX)
       COMMON/SEED/IX,KK
       SD=SQRT(VAR)
       PI=3.1415926
       IF(KK.EQ.1) GOTO 10
       RONE=RAND(IX)
       RTWO=RAND(IX)
       ZONE=SQRT(-2*ALOG(RONE))*COS(2*PI*RTWO)
       ZTWO=SQRT(-2*ALOG(RONE))*SIN(2*PI*RTWO)
       EX=ZONE*SD+RMEAN
       KK=1
       GOTO 15
    10 EX=ZTWO*SD+RMEAN
       KK=0
    15 RETURN
       END
```

```
C*********************************************************************C
C***                      FUNCTION RANDOM                          ***C
C*********************************************************************C
      FUNCTION RAND(IX)
      IX = IX*16807
      IF (IX.LT.0) IX = IX+2147483647+1
      RAND = IX
      RAND = RAND*0.465661E-9
      RETURN
      END
C*********************************************************************C
C***              SUBROUTINE UNIFORM DISTRIBUTION                  ***C
C*********************************************************************C
      SUBROUTINE UNIFOR(A,B,EX)
      COMMON/SEED/IX,KK
      YFL = RAND(IX)
      EX = ((B-A)*YFL) + A
      RETURN
      END
C*********************************************************************C
C***         SUBROUTINE TRUNCATED NORMAL DISTRIBUTION              ***C
C*********************************************************************C
      SUBROUTINE TNORM(XO,RM,S1,EX)
      DOUBLE PRECISION PROB,IFAULT
      COMMON/SEED/IX,KK
      S = SQRT(S1)
      ZO =(XO-RM)/S
      YFL = RAND(IX)
      CALL PP(ZO,PO)
      PROB = PO + YFL*(1-PO)
      IFAULT = 0.
      ZZ = VNORM(PROB,IFAULT)
      EX = (ZZ*S) + RM
      RETURN
      END
```

```
C*********************************************************************C
C***             SUBROUTINE TRUNCATED WEIBULL DISTRIBUTION           ***C
C*********************************************************************C
      SUBROUTINE TWEI(X0,WAL1,WBE1,EX)
      COMMON/SEED/IX,KK
      FX0 = 1 - ( EXP(-WAL1*(X0**WBE1)) )
      YFL = RAND(IX)
      WPRO = FX0 + ( (1-FX0)*YFL )
      B1 = 1/WBE1
      B2 = -( ALOG(1-WPRO))
      EX = (B2/WAL1)**B1
      RETURN
      END
C********************************************************************C
C***                       SUBROUTINE FACTORIAL                    ***C
C********************************************************************C
      SUBROUTINE FAC(N,GAM)
      IF ( (N .EQ. 0) .OR. (N .EQ. 1) ) THEN
         GAM = 1
         GOTO 20
      ENDIF
      GAM = 1
      DO 10 I = 2,N
      GAM = I*GAM
   10 CONTINUE
   20 RETURN
      END
C*********************************************************************C
C***                        SUBROUTINE SORT                          ***C
C*********************************************************************C
      SUBROUTINE SORT(EM,IS,EM1)
      DIMENSION EM(400),EM1(400)
      DO 10 I=1,IS
      EM1(I)=EM(I)
   10 CONTINUE
      K=IS-1
```

```
      DO 5 I=1,K
      K1=I+1
      DO 5 J=K1,IS
      IF(EM1(I).LE.EM1(J)) GOTO 5
      S=EM1(I)
      EM1(I)=EM1(J)
      EM1(J)=S
    5 CONTINUE
      RETURN
      END
C*********************************************************************C
C***              COMPUTE PROBABILITY OF NORMAL MODEL              ***C
C*********************************************************************C
      SUBROUTINE PP(ZZ,PZ)
            R = EXP(-ZZ*ZZ/2.)/2.5066282746
            WW = 1./(1.+.33267*ABS(ZZ))
            P = 1.-R*(.4361836*WW-.1201676*(WW**2)+.937298*(WW**3))
         IF (ZZ.GE.0) THEN
            PZ=P
         ELSE
            PZ=1.-P
         ENDIF
      RETURN
      END
C*********************************************************************C
C*** FUNCTION THE INVERSE OF THE CDF OF THE NORMAL DISTRIBUTION    ***C
C*********************************************************************C
      FUNCTION VNORM(PHI,IFAULT)
      DOUBLE PRECISION PHI
     *   ,IFAULT,PLIM,P0,P1,P2,P3,P4,Q0,Q1,Q2,Q3,Q4,P,VTEMP
C VNORM RETURNS THE INVERSE OF THE CDF OF THE NORMAL DISTRIBUTION
C IT USES A RATIONAL APPROXIMATION WHICH SEEMS TO HAVE A RELATIVE
C ACCURACY OF ABOUT 5 DECIMAL PLACES.
C REF.: DENNEDY AND GENTLE, STATISTICAL COMPUTING,DEKKER,1980.
C INPUT:
C PHI = PROBABILITY,0 <=  PHI <= 1.
C OUTPUTS:
```

```
C F INVERSE OF PHI, I.E., A VALUE SUCH THAT
C PROB(X <= VNORM) = PHI.
C IFAULT = 6 IF PHI OUT OF RANGE,ELSE 0
      DATA PLIM /1.0D-18/
      DATA P0/-0.322232431088D0/,P1/-1.0/,P2/-0.342242088547D0/
      DATA P3 / -0.0204231210245D0/, P4/-0.453642210148D-4/
      DATA Q0/0.099348462606D0/, Q1/0.588581570495D0/
      DATA Q2/0.531103462366D0/,Q3/0.10353775285D0/
      DATA Q4/0.38560700634D-2/
      IFAULT = 0
      P = PHI
      IF (P .GT. 0.5) P = 1. -P
      IF (P .GE. PLIM) GOTO 100
C THIS IS AS FAR OUT IN THE TAILS AS WE GO
      VTEMP = 8.
C CHECK FOR INPUT ERROR
      IF (P .LT. 0.) GOTO 9000
      GOTO 200
  100 Y = DSQRT(-DLOG(P*P))
      VTEMP = Y + ((((Y*P4 + P3)*Y+P2)*Y+P1)*Y+P0)/
     *     ((((Y*Q4+Q3)*Y+Q2)*Y+Q1)*Y+Q0)
  200 IF (PHI .LT. 0.5) VTEMP = -VTEMP
      VNORM = VTEMP
      RETURN
 9000 IFAULT = 6
      RETURN
      END
```

ประวัติผู้เขียน

นางสาวจงดี โรจนประศาสน์ เกิดเมื่อวันที่ 20 มีนาคม 2499 สำเร็จการศึกษาปริญญาบริหารธุรกิจบัณฑิต (บธ.บ.) จากภาควิชาธุรกิจศึกษา คณะบริหารธุรกิจ สถาบันเทคโนโลยีราชมงคล ในปีการศึกษา 2524 และเข้าศึกษาต่อในหลักสูตรสถิติศาสตรมหาบัณฑิต ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2533 ปัจจุบันรับราชการในตำแหน่งนักวิชาการเงินและบัญชี 5 คณะมนุษยศาสตร์และสังคมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตปัตตานี