รายการอ้างอิง

ภาษาไทย

ธีระพร วีระถาวร. ความน่าจะเป็นกับการประยุกต์. กรุงเทพมหานคร : สำนักพิมพ์อักษรกราฟฟิค, 2537

วีนัส พีชวณิชย์. ทฤษฎีความน่าจะเป็นและการประยุกต์. พิมพ์ครั้งที่ 6. กรุงเทพมหานคร:สำนักพิมพ์ประการพรึก,2535

สรชัย พิศาลบุตร. สถิติธุรกิจ. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : บริษัทวิทยพัฒน์ จำกัด, 2540

ภาษาต่างประเทศ

Andrews,David F.,Bickel,Peter J.,Hampel, F.R., Huber, Peter J., Rogers, William H., and Tukey, John W. Robust Estimates of Location: Survey and Advances, Princeton, N.J.:Princeton University Press. 1972

Averill M.Law and W.David Kelton. Simulation modeling and Analysis. 2nd . Edition. New York: Mc Graw-Hill , Inc., 1991

Box, George E.P., and Anderson, Sigurd L. 'Permutation Theory in the Derivation of Robust Criteria and the Study of Departures from Assumption,' Journal of the Royal Statistical Society, Ser. B (1955) :1-26

Efron,B. and Robert J.T. An introduction to the Bootstrap. New York: Chapman & Hall ,1993

Evans,M.,Hastings,N.,Peacock,B. Statistical Distribution. 2nd.Edition. New York : John Wiley &Sons,1993

Hartigan, John A. 'Using Subsample Values as Typical Values,' Journal of the American Statistical Association (1969):1303-1317

Johnson, N.J. 'Modified t Tests and Confidence Intervals for Asymmetrical Populations,' Journal of the American Statistical Association (1978):536-544

Mood, A.M., Graybill, F.A., and Boes, D.C. Introduction to the Theory of Statistics. New York: John Wiley,1974

Olkin,L., Leon,J.G., and Cyrus, D. Probability Models and Applications. New York: Macmillan Publishing Co.,Inc.,1980

รายการอ้างอิง(ต่อ)

Sutton, C.D. 'Computer-Intensive Methods for Tests About the Mean of an Asymmetrical Distribution,' Journal of the American Statistical Association (1993):802-810

Tukey, John W., Data Analysis and Behavioral Science unpublished manuscript.(1964)

Yune, Karen K. 'The Two-Sample Trimmed t for Unequal Population Variances,' biometrika (1974):165-170

ภาคผนวก ก

โปรแกรมสำหรับคำนวณค่าความน่าจะเป็นของความคลาดเคลื่อนประเภทที่ 1 และอำนาจการทดสอบของสถิติทดสอบ 4 วิธี

```
C  ***********************************************
C  *******    MAIN PROGRAM              **********
C  ***********************************************
      DOUBLE PRECISION X,XX,T,MUE,XY,B,XBAR1,SD1,ALPHA,BETA
      COMMON /DATA/X(5000)
     *       /CONST/MUE
     *       /T_STU/STUD_T
     *       /POWER/VCONS
     *       /SEED/IX
      DIMENSION  XY(5000),B(5000)
      DATA ITS1,ITS2,ITS3/0,0,0/
      DATA ITJS1,ITJS2,ITJS3/0,0,0/
      DATA ILC1,ILC2,ILC3/0,0,0/
      DATA ISUT1,ISUT2,ISUT3/0,0,0/
      OPEN(6,FILE='C:\FORTRAN\OUTPUT.OUT')
      IX = 973253
      N = 10
      VK = 0.0
      SIGNIFICANT = 0.01
      LOOP1 = 5000
      ALPHA = 0.8630
      BETA =  1.0000
      DMEAN = 0.0000
      SIGMA = 0.0831
      TS1  = 0
      TS2  = 0
      TS3  = 0
      TJS1 = 0
      TJS2 = 0
      TJS3 = 0
      TLC1 = 0
      TLC2 = 0
```

```
      TLC3 = 0
      TSU1 = 0
      TSU2 = 0
      TSU3 = 0
      ITS1 = 0
      ITS2 = 0
      ITS3 = 0
      ITJS1 = 0
      ITJS2 = 0
      ITJS3 = 0
      ILC1 = 0
      ILC2 = 0
      ILC3 = 0
      ISUT1 = 0
      ISUT2 = 0
      ISUT3 = 0
C     ******************************************************
C     *              MEAN AND STANDARDIVATION              *
C     *                   OF POPULATION                    *
C     ******************************************************
C     *              GAMMA  DISTRIBUTION                   *
C     ******************************************************
        SDP = SQRT(ALPHA*(BETA**2))
        VCONS = VK*(SDP/SQRT(N))
        MUE = (ALPHA*BETA)
C     ********************************************
C     *      LOG-NORMAL DISTRIBUTION             *
C     ********************************************
        ZM = (EXP(DMEAN))**2
        ZW = EXP(SIGMA**2)
        WW = ZW-1
        SDP = SQRT(ZM*ZW*WW)
        VCONS = VK*(SDP/SQRT(N))
        MUE = (EXP(DMEAN+(SIGMA**2)/2))
```

```
C     ********************************************************
C     *                  WEIBULL DISTRIBUTION                *
C     ********************************************************
C     *       BECAUSE MEAN = BETA*FG(1+(1/ALPHA))            *
C     *       AND     SD = BETA*SQRT(FG(1+(2/ALPHA))         *
C     *                       -FG(1+(1/ALPHA))**2)           *
C     *    SO THAT   WC = FG(1+(1/ALPHA))                    *
C     *                CONS = SQRT(FG(1+(2/ALPHA))           *
C     *                       -FG(1+(1/ALPHA))**2)           *
C     *             IF    FG = GAMMA FUNCTION                *
C     ********************************************************
        MUE = 1.0779
        GFSD= 1.2533
        SDP = BETA*GFSD
        VCONS = VK*(SDP/SQRT(N))
C*******************************************************
        DO 50 J = 1,LOOP1
        DO 10 I = 1,N
        ALPHA = 0.8630
        BETA  = 1.0000
        DMEAN = 0.0000
        SIGMA = 0.0831
        CALL GAM_1(ALPHA,XX)
        CALL GAM_2(ALPHA,XX)
        CALL GAM_3(BETA,XX)
        CALL WEI(ALPHA,BETA,XX)
        CALL LOGN(DMEAN,SIGMA,IK,XX)
        X(I) = XX
        XY(I) = XX
        B(I) = X(I)+VCONS
10      CONTINUE
        CALL XBAR_SD(XBAR1,SD1,N,B)
        CALL STUDENT(T,N,XBAR1,SD1)
        STUD_T = T
        CALL CRI_TSD(STUD_T,ITS1,ITS2,ITS3)
```

```
      CALL JOHNSON(T_JOHN,N,B,XBAR1,SD1,SXXX1)
      T1 = T_JOHN
      CALL CRI_TJH(T1,ITJS1,ITJS2,ITJS3)
      CALL L_CHEN(T_CHEN,STUD_T,N,SD1,SXXX1)
      CALL CRI_TLC(T_CHEN,ILC1,ILC2,ILC3)
      CALL SUT(T_SUT1,T_SUT,N,T1,XY)
      CALL CRI_SUT(T_SUT1,T_SUT,ISUT1,ISUT2,ISUT3,T1)
50    CONTINUE
      TS1 = FLOAT(ITS1)/LOOP1
      TS2 = FLOAT(ITS2)/LOOP1
      TS3 = FLOAT(ITS3)/LOOP1
      TJS1 = FLOAT(ITJS1)/LOOP1
      TJS2 = FLOAT(ITJS2)/LOOP1
      TJS3 = FLOAT(ITJS3)/LOOP1
      TLC1 = FLOAT(ILC1)/LOOP1
      TLC2 = FLOAT(ILC2)/LOOP1
      TLC3 = FLOAT(ILC3)/LOOP1
      TSU1 = FLOAT(ISUT1)/LOOP1
      TSU2 = FLOAT(ISUT2)/LOOP1
      TSU3 = FLOAT(ISUT3)/LOOP1
C     WRITE(6,*) '      ***************************'
      WRITE(6,60) TS1,TS2,TS3
60    FORMAT(8X,'TS1 = ',F6.4,6X,'TS2 = ',F6.4,6X,'TS3 = ',F6.4)
      WRITE(6,70) TJS1,TJS2,TJS3
70    FORMAT(&x,'TJS1 = ',F6.4,6X,'TJS2 = ',F6.4,6X,'TJS3 = ',F6.4)
      WRITE(6,120) TLC1,TLC2,TLC3
120   FORMAT(8X,'TLC1 = ',F6.4,6X,'TLC2 = ',F6.4,6X,'TLC3 = ',F6.4)
      WRITE(6,130) TSU1,TSU2,TSU3
130   FORMAT(8X,'TSU1 = ',F6.4,6X,'TSU2 = ',F6.4,6X,'TSU3 = ',F6.4)
      WRITE(6,*) '      ###########################'
      STOP
      END
```

```
C  ************************************************
C  ********  GENERATE  RANDOM NUMBER   ********
C  ************************************************
      SUBROUTINE RAND(IX,YFL)
      REAL YFL
      IY = IX*16807
      IF (IY.LT.0) IY = IY+2147483647+1
      YFL = IY
      YFL = YFL*0.465661E-9
      IX = IY
      RETURN
      END
C  ************************************************
C  *****  GAMMA DISTRIBUTION : 0 < ALPHA < 1  *****
C  ************************************************
      SUBROUTINE GAM_1(ALPHA,XX)
      DOUBLE PRECISION  XX,ALPHA,B,P,Y,CHK1,CHK2
      COMMON /SEED/IX
      B = (EXP(1)+ALPHA)/EXP(1)
10    CALL RAND(IX,YFL)
      RONE = YFL
      P = B*RONE
      CALL RAND(IX,YFL)
      RTWO = YFL
      IF (P.GT.1.00) GOTO 40
            Y = P**(1.00/ALPHA)
            CHK1 = EXP(-Y)
      IF (RTWO.LE.CHK1) THEN
            XX = Y
      ELSE
            GOTO 10
      END IF
      RETURN
40    Y = -ALOG((B-P)/ALPHA)
      CHK2 = Y**(ALPHA-1.00)
```

```
      IF (RTWO.LE.CHK2) THEN
            XX=Y
      ELSE
            GOTO 10
      END IF
      RETURN
      END
C     ***************************************************************
C     *****        GAMMA DISTRIBUTION :  ALPHA > 1          *****
C     ***************************************************************
      SUBROUTINE GAM_2(ALPHA,XX)
      DOUBLE PRECISION  XX,ALPHA
      COMMON /SEED/IX
      A1 = 1/SQRT((2*ALPHA)-1)
      A2 = ALPHA-ALOG(4)
      A3 = ALPHA+(1/A1)
      A4 = 1+ALOG(4.5)
10    CALL RAND(IX,YFL)
      RAN1 = YFL
      CALL RAND(IX,YFL)
      RAN2 = YFL
      V = A1*ALOG(RAN1/(1-RAN1))
      Y = ALPHA*EXP(V)
      Z = (RAN1**2)*RAN2
      W = A2+(A3*V)-Y
      CHK1 = W+(A4)-(4.5*Z)
      IF (CHK1.GE.0) THEN
            XX = Y
      RETURN
      END IF
      CHK2 = ALOG(Z)
      IF (W.GE.CHK2) THEN
            XX = Y
      ELSE
            GOTO 10
```

```
      END IF
      RETURN
      END
C  ************************************************************
C  ***          GAMMA DISTRIBUTION : ALPHA = 1              ***
C  ************************************************************
      SUBROUTINE GAM_3(BETA,XX)
      DOUBLE PRECISION  XX,BETA
      COMMON /SEED/IX
10    CALL RAND(IX,YFL)
      V = -ALOG(YFL)
      XX = BETA * V
      RETURN
      END
C  ************************************************
C  ************  WEIBULL DISTRIBUTION  ************
C  ************************************************
      SUBROUTINE WEI(ALPHA,BETA,XX)
      DOUBLE PRECISION  XX,ALPHA,BETA
      COMMON /SEED/IX
      CALL RAND(IX,YFL)
      AK = -ALOG(1-YFL)
      WX = 1.0/ALPHA
      XX = BETA*(AK**WX)
      RETURN
      END
C  ************************************************
C  ************  NORMAL DISTRIBUTION  ************
C  ************************************************
      SUBROUTINE NORM(DMEAN,SIGMA,IK,XL)
      DOUBLE PRECISION  XL
      COMMON /SEED/IX
      PI = 3.1415926
      IF (IK.EQ.1) GOTO 10
5     CALL RAND(IX,YFL)
```

```
      IF ((YFL.LE.0.).OR.(YFL.GT.1.)) GOTO 5
            RONE = YFL
6     CALL RAND(IX,YFL)
      IF ((YFL.LE.0.).OR.(YFL.GT.1.)) GOTO 6
            RTWO = YFL
            ZONE = SQRT(-2*ALOG(RONE))*COS(2*PI*RTWO)
            ZTWO=SQRT(-2*ALOG(RONE))*SIN(2*PI*RTWO)
            XL  = ZONE * SIGMA + DMEAN
      IK = 1
      RETURN
10    XL  = ZTWO * SIGMA + DMEAN
      IK = 0
      RETURN
      END
C  ***************************************************
C  **********  LOGNORMAL DISTRIBUTION  **********
C  ***************************************************
      SUBROUTINE LOGN(DMEAN,SIGMA,IK,XX)
      DOUBLE PRECISION XX,X,XL
      COMMON /SEED/IX
*     /DATA/X(5000)
      CALL NORM(DMEAN,SIGMA,IK,XL)
      XX = EXP(XL)
      RETURN
      END

C  ***************************************************************************
C  **********  CALCULATE X_BAR AND STANDARD DIVATION  **********
C  ***************************************************************************
      SUBROUTINE XBAR_SD(XBAR1,SD1,N,B)
      DOUBLE PRECISION X,SX,SXX,SXX1,XBAR1,SD1,VAR,B
      COMMON /DATA/X(5000)
      DIMENSION B(5000)
      SX = 0.00
      SXX = 0.00
```

```
      SXX1 = 0.00
      DO 10 I=1,N
      SX = SX+B(I)
10    CONTINUE
      XBAR = SX/N
      XBAR1 = XBAR
      DO 20 I=1,N
      SXX = (B(I)-XBAR1)**2
      SXX1 = SXX1+SXX
20    CONTINUE
      VAR = (SXX1)/(N-1)
      SD  = SQRT(VAR)
      SD1 = SD
      RETURN
       END
C  ***************************************************
C  *********  CALCULATE STUDENT  T  TEST  *********
C  ***************************************************
      SUBROUTINE STUDENT(T,N,XBAR1,SD1)
      DOUBLE PRECISION TN,SN,T,MUE,XBAR1,SD1
      COMMON  /CONST/MUE
      TN = N
      SN = SD1/SQRT(TN)
      T = (XBAR1-MUE)/SN
       RETURN
      END
C  ***************************************************
C  *******  CALCULATE JOHNSON'S  T  TEST  **********
C  ***************************************************
      SUBROUTINE JOHNSON(T_JOHN,N,B,XBAR1,SD1,SXXX1)
      DOUBLE PRECISION MUE,B,XBAR1,SD1
      COMMON /CONST/MUE
      DIMENSION B(5000)
      SXXX = 0.0
      SXXX1 = 0.0
```

```
      DO 10 I=1,N
      SXXX = (B(I)-XBAR1)**3
      SXXX1 = SXXX1+SXXX
10    CONTINUE
      TMOMENT = SXXX1/(N)
      TJ1 = XBAR1-MUE
      TJ2 = TMOMENT/(6*(SD1**2)*N)
      TJ3 = TMOMENT*(TJ1**2)/(3*(SD1**4))
      TJ4 = SQRT((SD1)**2/(N))
      T_JOHN = (TJ1+TJ2+TJ3)/TJ4
      RETURN
      END
C     ************************************************
C     *****  CALCULATE LING CHEN'S T TEST  *********
C     ************************************************
      SUBROUTINE L_CHEN(T_CHEN,STUD_T,N,SD1,SXXX1)
      DOUBLE PRECISION X,SD1
      COMMON /DATA/X(5000)
      REAL LC1,LC2
      SKEW=(SXXX1*N)/((N-1)*(N-2)*(SD1**3))
      LC1 = SKEW*(1+2*(STUD_T**2))/(6*SQRT(N))
      LC2 = (SKEW**2)*(STUD_T+2*(STUD_T**2))/(9*N)
      T_CHEN = STUD_T+LC1+LC2
      RETURN
      END
C     ************************************************
C     ********  CALCULATE BOOTSTRAP   ************
C     ************************************************
      SUBROUTINE B_STAP(N,XY,T_JHON)
      DOUBLE PRECISION PP,XY,B,XBAR1,SD1
      COMMON /POWER/VCONS
*                 /SEED/IX
      DIMENSION PP(5000),XY(5000),B(5000)
      DO 11 I=1,N
      PP(I) = FLOAT(I)/FLOAT(N)
```

```
11      CONTINUE
        DO 20 J = 1,N
        CALL RAND(IX,YFL)
        DO 10 IT =1,N
        IT1 = IT-1
        IF (IT1.EQ.0) THEN
                A1 = 0.0
        ELSE
                A1 = PP(IT1)
        END IF
                A2 = PP(IT)
        IF ((YFL.GT.A1).AND.(YFL.LE.A2)) THEN
                B(J)  = XY(IT)+VCONS
        END IF
10      CONTINUE
20      CONTINUE
        CALL XBAR_SD(XBAR1,SD1,N,B)
        CALL JHONSON(T_JHON,N,B,XBAR1,SD1,SXXX1)
        RETURN
        END
C   ************************************************
C   ***** CALCULATE SUTTON'S COMPOSITE TEST  ***
C   ************************************************
        SUBROUTINE SUT(T_SUT1,T_SUT,N,T1,XY)
        DOUBLE PRECISION XY,MUE,XBAR_T,VAR_T,SD_T,SXX1,SXX
        COMMON /CONST/MUE
*                /POWER/VCONS
*                /SEED/IX
        DIMENSION TT(5000),XY(5000)
        LOOP = 120
        DO 30 I=1,LOOP
        CALL B_STAP(N,XY,T_JHON)
        TT(I) = T_JHON
30      CONTINUE
        SX = 0.0
```

```
      SXX = 0.0
      SXX1 = 0.0
      DO 110 I=1,LOOP
      SX = SX+TT(I)
110   CONTINUE
      XBAR_T = SX/LOOP
      DO 150 I=1,LOOP
      SXX = (TT(I)-XBAR_T)**2
      SXX1 = SXX1+SXX
150   CONTINUE
      VAR_T = (SXX1)/(LOOP-1)
      SD_T  = SQRT(VAR_T)
      T_SUT1 = T1/SD_T
      AA = N-1
      BB = N-3
      TSUT = AA/BB
      T_SUT = (SQRT(TSUT))*(T_SUT1)
      RETURN
      END
C  ***********************************************
C  *****       CRITICAL STUDENT                ***
C  ***********************************************
       SUBROUTINE CRI_TSD(STUD_T,ITS1,ITS2,ITS3)
       INTEGER ITS1,ITS2,ITS3
       REAL LEFT,RIGT,HALT1,HALT2
C  *************** 90% ****************
       LEFT  = -1.383
       RIGT  =  1.383
       HALT1 = -1.833
       HALT2 =  1.833
C  *************** 95% ****************
       LEFT  = -1.833
       RIGT  =  1.833
       HALT1 = -2.262
       HALT2 =  2.262
```

```
C  ********** 99% **********
      LEFT  = -2.821
      RIGT  =  2.821
      HALT1 = -3.250
      HALT2 =  3.250
      IF (STUD_T.LT.LEFT) ITS1 = ITS1+1
      IF (STUD_T.GT.RIGT) ITS2 = ITS2+1
      IF ((STUD_T.LT.HALT1).OR.(STUD_T.GT.HALT2)) ITS3 = ITS3+1
      RETURN
      END
C  ***********************************************
C  *****      CRITICAL JHONSON                 ***
C  ***********************************************
      SUBROUTINE CRI_TJH(T1,ITJS1,ITJS2,ITJS3)
      INTEGER ITJS1,ITJS2,ITJS3
      REAL LEFT,RIGT,HALT1,HALT2
C  ********** 90% **********
      LEFT  = -1.383
      RIGT  =  1.383
      HALT1 = -1.833
      HALT2 =  1.833
C  ********** 95% **********
      LEFT  = -1.833
      RIGT  =  1.833
      HALT1 = -2.262
      HALT2 =  2.262
C  ********** 99% **********
      LEFT  = -2.821
      RIGT  =  2.821
      HALT1 = -3.250
      HALT2 =  3.250
      IF (T1.LT.LEFT) ITJS1 = ITJS1+1
      IF (T1.GT.RIGT) ITJS2 = ITJS2+1
      IF ((T1.LT.HALT1).OR.(T1.GT.HALT2)) THEN
            ITJS3 = ITJS3+1
```

```
      ENDIF
      RETURN
      END
C  ***********************************************
C  *****      CRITICAL LING CHEN              *****
C  ***********************************************
      SUBROUTINE CRI_TLC(T_CHEN,ILC1,ILC2,ILC3)
      INTEGER ILC1,ILC2,ILC3
      REAL LEFTZ,RIGTZ,HALTZ1,HALTZ2
C  *************** 90% ****************
      LEFTZ = -1.332
      RIGTZ =  1.332
      HALTZ1 = -1.739
      HALTZ2 =  1.739
C  *************** 95% ****************
      LEFTZ = -1.739
      RIGTZ =  1.739
      HALTZ1 = -2.111
      HALTZ2 =  2.111
C  *************** 99% ****************
      LEFTZ = -2.574
      RIGTZ =  2.574
      HALTZ1 = -2.913
      HALTZ2 =  2.913
      IF (T_CHEN.LT.LEFTZ) ILC1 = ILC1+1
      IF (T_CHEN.GT.RIGTZ) ILC2 = ILC2+1
      IF ((T_CHEN.LT.HALTZ1).OR.(T_CHEN.GT.HALTZ2)) THEN
            ILC3 = ILC3+1
      ENDIF
      RETURN
      END
```

```
C  ***********************************************
C  *****      CRITICAL SUTTON                 *****
C  ***********************************************
      SUBROUTINE CRI_SUT(T_SUT1,T_SUT,ISUT1,ISUT2,ISUT3,T1)
      INTEGER ISUT1,ISUT2,ISUT3
      REAL  LEFT,RIGT,HALT1,HALT2,LEFZ,RIGZ,HALZ1,HALZ2
C  *************** 90% ****************
         LEFT  = -1.383
         RIGT  =  1.383
         HALT1 = -1.833
         HALT2 =  1.833
         LEFZ  = -1.282
         RIGZ  =  1.282
         HALZ1 = -1.645
         HALZ2 =  1.645
C  *************** 95% ****************
         LEFT  = -1.833
         RIGT  =  1.833
         HALT1 = -2.262
         HALT2 =  2.262
         LEFZ  = -1.645
         RIGZ  =  1.645
         HALZ1 = -1.960
         HALZ2 =  1.960
C  *************** 99% ****************
         LEFT  = -2.821
         RIGT  =  2.821
         HALT1 = -3.250
         HALT2 =  3.250
         LEFZ  = -2.327
         RIGZ  =  2.327
         HALZ1 = -2.575
         HALZ2 =  2.575
IF ((T1.LT.LEFT).or.((T_SUT1.LT.LEFZ).OR.(T_SUT.LT.LEFT)))
*        ISUT1 = ISUT1+1
```

```
      IF ((T1.GT.RIGT).or.((T_SUT1.GT.RIGZ).OR.(T_SUT.GT.RIGT)))
     *     ISUT2 = ISUT2+1
      IF (((T1.LT.HALT1).OR.(T1.GT.HALT2)).or.
     *     (((T_SUT1.LT.HALZ1).OR.(T_SUT1.GT.HALZ2)).OR.
     *     ((T_SUT.LT.HALT1).OR.(T_SUT.GT.HALT2)))) THEN
           ISUT3 = ISUT3+1
           ENDIF
           RETURN
      END
```

ภาคผนวก ข

การประมาณค่าพารามิเตอร์โดยวิธีบูตสแตรป (Bootstrap Method)

วิธีหาตัวประมาณของพารามิเตอร์โดยวิธีบูตสแตรปนี้ เป็นวิธีที่เสนอขึ้นโดยแบรดเลย์ เอฟรอน (Bradley Efron) ในปีค.ศ.1979 โดยมีหลักเกณฑ์ คือ จะทำการสุ่มข้อมูลเท่ากับจำนวนขนาดตัวอย่างจากข้อมูลที่เก็บรวบรวมมาแบบใส่คืน (with replacement) เพื่อสร้างข้อมูลชุดใหม่ แล้วนำมาใช้ในการประมาณค่าพารามิเตอร์ที่สนใจ

การหาตัวประมาณด้วยวิธีนี้ มีขั้นตอนการทำงานดังนี้

1. สร้างตัวเลขสุ่มที่มีค่าอยู่ระหว่าง 0 กับ 1 เพื่อนำไปใช้ในการสุ่มตัวอย่างแบบใส่คืน
2. จากตัวอย่างที่สุ่มได้แต่ละชุด นำมาหาค่าสถิติหรือค่าประมาณของพารามิเตอร์ที่สนใจ

<u>การสุ่มตัวอย่างแบบใส่คืน</u> (Sampling with replacement)

เป็นการสุ่มตัวอย่าง ออกมากจากประชากรทีละตัวอย่าง โดยที่หน่วยตัวอย่างที่ถูกเลือกมาแล้วมีโอกาสถูกเลือกซ้ำได้อีก นั่นคือ หน่วยตัวอย่างมีโอกาส (Probability) ในการถูกสุ่มแต่ละครั้งเท่ากันเท่ากับ 1/N เมื่อ N คือ ขนาดของประชากร

งานวิจัยนี้ได้ใช้คอมพิวเตอร์เป็นเครื่องมือช่วยในการสุ่มตัวอย่างแบบใส่คืน โดยใช้ตัวอย่างที่มีการแจกแจงแบบสม่ำเสมอ ที่มีค่าอยู่ในช่วง [0,1] เป็นตัวเปรียบเทียบกับค่าความน่าจะเป็นสะสม เพื่อกำหนดหน่วยตัวอย่าง ตามจำนวนที่ต้องการ จึงมีขั้นตอนในการสุ่มดังนี้

1. คำนวณหาความน่าจะเป็นของแต่ละหน่วยตัวอย่าง ซึ่งเท่ากับ 1/N
2. หาค่าความน่าจะเป็นสะสม แล้วจัดเป็นช่วง
3. สร้างตัวเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอซึ่งมีค่าอยู่ในช่วง [0,1]
4. นำตัวเลขสุ่มที่ได้ในข้อ 3 มาเปรียบเทียบกับค่าความน่าจะเป็นสะสม ถ้าตัวเลขที่สุ่มได้ตกอยู่ในช่วงใด หน่วยนั้นจะถูกเลือกมาเป็นตัวอย่าง
5. ดำเนินการตามขั้นตอนที่ 3 และ 4 จำนวน n ครั้ง เมื่อ n คือ ขนาดตัวอย่างที่ต้องการ

ตัวอย่าง การสุ่มตัวอย่างแบบใส่คืน เมื่อ N = 10 และ n = 5

คำนวณหาความน่าจะเป็นของแต่ละหน่วยตัวอย่าง เท่ากับ 0.1 และสามารถนำมาสร้างตาราง การแจกแจงสะสมได้ดังนี้

หน่วยตัวอย่าง	ความน่าจะเป็น	ความน่าจะเป็นสะสม	ช่วงความน่าจะเป็นสะสม
1	0.1	0.1	0.01 – 0.10
2	0.1	0.2	0.11 – 0.20
3	0.1	0.3	0.21 – 0.30
4	0.1	0.4	0.31 – 0.40
5	0.1	0.5	0.41 – 0.50
6	0.1	0.6	0.51 – 0.60
7	0.1	0.7	0.61 – 0.70
8	0.1	0.8	0.71 – 0.80
9	0.1	0.9	0.81 – 0.90
10	0.1	0.10	0.91 – 1.00

สมมติ เลขสุ่มตัวที่ 1 มีค่า 0.48 หน่วยที่ 5 จะถูกเลือกเป็นตัวอย่าง
เลขสุ่มตัวที่ 2 มีค่า 0.24 หน่วยที่ 3 จะถูกเลือกเป็นตัวอย่าง
เลขสุ่มตัวที่ 3 มีค่า 0.61 หน่วยที่ 7 จะถูกเลือกเป็นตัวอย่าง
เลขสุ่มตัวที่ 4 มีค่า 0.30 หน่วยที่ 3 จะถูกเลือกเป็นตัวอย่าง
เลขสุ่มตัวที่ 5 มีค่า 0.96 หน่วยที่ 10 จะถูกเลือกเป็นตัวอย่าง

จะเห็นว่าหน่วยตัวอย่าง มีโอกาสถูกเลือกได้มากกว่า 1 ครั้ง ขึ้นอยู่กับค่าของเลขสุ่มว่าตกอยู่ในช่วงใดของความน่าจะเป็นสะสม

ประวัติผู้วิจัย

นางสาวอัญชนา ลีลาจรัสกุล เกิดวันที่ 12 กันยายน 2514 สำเร็จการศึกษาปริญญาวิทยาศาสตร์บัณฑิต(วท.บ.) สาขาสถิติประยุกต์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้า พระนครเหนือ ปีการศึกษา 2536 และเข้าศึกษาต่อในหลักสูตรสถิติศาสตร์มหาบัณฑิต ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2538