

บรรณานุกรม

ภาษาไทย

- เอกชัย ชัยประเสริฐวิสิทธิ์. การวิเคราะห์สหสัมพันธ์และการถดถอย. กรุงเทพมหานคร : สาขาวิชาสถิติ คณะศิลปศาสตร์ มหาวิทยาลัยธรรมศาสตร์, 2525
- สมจิตร วัฒนาชยากุล. สถิติวิเคราะห์เบื้องต้น. กรุงเทพมหานคร : สาขาวิชาคณิตศาสตร์-สถิติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์, 2529
- ศิริชัย กาญจนราสี. สถิติศาสตร์ หลักการและเหตุผล. กรุงเทพมหานคร : ภาควิชาวิจัยการศึกษา ครุศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2526.
- อภิชาติ พงษ์ศรีหุลชัย. การสุ่มตัวอย่าง. กรุงเทพมหานคร : สำนักงานเศรษฐกิจการเกษตร กระทรวงเกษตรและสหกรณ์, 2530
- สุชาดา กิรินทร์. การสำรวจตัวอย่าง. กรุงเทพมหานคร : ภาควิชาสถิติ จุฬาลงกรณ์มหาวิทยาลัย, 2525.

ภาษาต่างประเทศ

- Donald F. Morrison. Multivariate Statistical Methods. Tokyo : Mcgraw Hill, 1978.
- Frederick F. Stephan And Philip J. McCarthy. Sampling Opions An Analysis Of Survey Produre. New York : John Wiley & Sons, 1958.
- William G. Cochran. Sampling Techniques. New York : John Wiley & Sons, 1958.



ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

{*****}
{**          THESIS          **}
{** COMPARATIVE STRTITY WITH QUOTA SAMPLING **}
{**      BY MR. PRACHA SUWATTANAPUNGUL      **}
{*****}
PROGRAM STRATIFY_QUOTA(INPUT,OUTPUT);  {** FACTOR1 **}
USES  CRT,PRINTER;
CONST
  I_COUNT   = 5000;    {TOTAL OBSERVATION IN EACH VARIABLE (N)}
  K_COUNT   = 12;     {TOTAL POPULATION}
  I_MEAN_NORM = 10000.00; {MEAN OF DATA Y = 10000} {CV = 5%,10%,30%,50%}
  S_COUNT   = 100;   {REPEAT RANDOM STRATIFIED OR SELLECT QUOTA}
TYPE
  VARXY      = RECORD
                X1,Y      : REAL;
            END;
  VARGR1     = RECORD
                GROUP1   : REAL;
            END;
  VARGR2     = RECORD
                GROUP2   : REAL;
            END;
  VARGR3     = RECORD
                GROUP3   : REAL;
            END;
  VARGR4     = RECORD
                GROUP4   : REAL;
            END;
  VARGR5     = RECORD
                GROUP5   : REAL;
            END;
  VARGR6     = RECORD
                GROUP6   : REAL;
            END;
  VARGR7     = RECORD
                GROUP7   : REAL;
            END;
  VARGR8     = RECORD
                GROUP8   : REAL;
            END;
  VARGR9     = RECORD
                GROUP9   : REAL;
            END;
  VARGR10    = RECORD
                GROUP10  : REAL;
            END;
  VARGR11    = RECORD
                GROUP11  : REAL;
            END;
  VARGR12    = RECORD
                GROUP12  : REAL;
            END;

```

```

END;
ONE_DIMENSION = ARRAY [1..I_COUNT] OF REAL;
ONE_INT       = ARRAY [1..I_COUNT] OF BYTE;
ONE_D2        = ARRAY [1..12]     OF INTEGER;
ONE_ISAM      = ARRAY [1..3]      OF INTEGER;
ONE_STD       = ARRAY [1..12]     OF REAL;
VAR
FILEXY        : FILE OF VARXY;
RECY          : VARXY;
FILEGR1       : FILE OF VARGR1;
RECGR1        : VARGR1;
FILEGR2       : FILE OF VARGR2;
RECGR2        : VARGR2;
FILEGR3       : FILE OF VARGR3;
RECGR3        : VARGR3;
FILEGR4       : FILE OF VARGR4;
RECGR4        : VARGR4;
FILEGR5       : FILE OF VARGR5;
RECGR5        : VARGR5;
FILEGR6       : FILE OF VARGR6;
RECGR6        : VARGR6;
FILEGR7       : FILE OF VARGR7;
RECGR7        : VARGR7;
FILEGR8       : FILE OF VARGR8;
RECGR8        : VARGR8;
FILEGR9       : FILE OF VARGR9;
RECGR9        : VARGR9;
FILEGR10      : FILE OF VARGR10;
RECGR10       : VARGR10;
FILEGR11      : FILE OF VARGR11;
RECGR11       : VARGR11;
FILEGR12      : FILE OF VARGR12;
RECGR12       : VARGR12;

I,J,K,L,M,N,KK : INTEGER;
G               : INTEGER;
NO_SAM          : ONE_D2;
NO,NOQ         : ONE_D2;
YG,X           : ONE_DIMENSION;
IX,IX1,IX2     : LONGINT;
MEAN_Y,MEAN_SAM : REAL;
MEAN_STRAT,MEAN_QUOTA,SUMR,R_MEAN : REAL;
ERROR_STRAT,ERROR_QUOTA : REAL;
STD_STRAT,STD_QUOTA : REAL;
ZTWO           : REAL;
CORR_X_Y       : ONE_STD;
  {DEGREE CORRELATION COEFFICIENT BETWEEN Y AND X = 0.1,0.5,0.9}
  {CV = 2%,5%,10%,20}
I_STD_NORM     : ONE_STD;
  {STANDART DIVATION OF DATA Y = 200,500,1000,2000}
I_SAM          : ONE_ISAM;
  {SAMPLE n = 250,500,750}
CVP            : REAL;
SUM_STDQT     : REAL;
SUM_STDST     : REAL;

```

```

[*****]
[**      ROUTINE SIMULATE      **]
[**  N O R M A L  D I S T R I B U T I O N  **]
[**          F O R          **]
[**  I N D E P E N D E N T  V A R I A B L E  ---> X1,X2  **]
[**  D E P E N D E N T  V A R I A B L E  ---> Y  **]
[*****]

```

```

[*****]
[**      SUBROUTINE FOR MAKING      **]
[**  R A N D O M  N U M B E R  **]
[*****]

```

```

FUNCTION RAND(IXX : LONGINT) : REAL;
BEGIN
    IX := IXX * 16807;    {65539}
    IF (IX < 0) THEN IX := IX + 2147483647 + 1;
    RAND := IX * 0.4656613E-9;
END;

```

```

[*****]
[** SUBPROGRAM TEST MULTICOLINEARITY **]
[**   OF SIMULATE DATA             **]
[*****]
PROCEDURE TEST_CORR;
VAR
    SUMXY,SUMX,SUMY,SUMXX,SUMYY,R : REAL ;
BEGIN
    R := 0.0;SUMXY := 0;SUMX := 0;SUMY := 0;SUMXX := 0;SUMYY := 0;
    ASSIGN (FILEXY,'XY1D01');RESET (FILEXY);
    WHILE NOT EOF(FILEXY) DO BEGIN
        READ (FILEXY,RECY);
        SUMXY := SUMXY + RECY.Y * RECY.X1;
        SUMXX := SUMXX + RECY.Y * RECY.Y;
        SUMYY := SUMYY + RECY.X1 * RECY.X1;
        SUMY := SUMY + RECY.X1; SUMX := SUMX + RECY.Y;
    END;
    CLOSE (FILEXY);
    R := I_COUNT * SUMXY - SUMX * SUMY;
    R := R/SQRT((I_COUNT*SUMXX-SQR(SUMX))*(I_COUNT*SUMYY-SQR(SUMY)));
    SUMR := SUMR + R;
    WRITELN('VALUE CORRELATION SETTING = ',CORR_X_Y[K]:5:3);
    WRITELN('CORREALTION BETWEEN X AND Y = ',R:5:3);
    WRITELN(LST);
    WRITELN(LST,'CORREALTION BETWEEN X AND Y = ',R:5:3);
    WRITELN(LST,'VALUE CORRELATION SETTING = ',CORR_X_Y[K]:5:3);
END;

```

```

{*****}
{**      SUBROUTINE FOR MAKING      **}
{**      N O R M A L ( 0 , 1 )      **}
{*****}

```

```
PROCEDURE NORMAL(VAR NORM :REAL);
```

```
CONST PI = 3.1415926;
```

```
VAR
```

```
  RONE,RTWO : REAL;
```

```
  ZONE      : REAL;
```

```
BEGIN
```

```
  IF KK = 1 THEN
```

```
  BEGIN
```

```
    NORM := ZTWO ;
```

```
    KK := 0;
```

```
  END
```

```
  ELSE
```

```
  BEGIN
```

```
    RONE := RAND(IX);
```

```
    RTWO := RAND(IX);
```

```
    ZONE := SQRT(-2*LN(RONE)) * COS(2*PI*RTWO);
```

```
    ZTWO := SQRT(-2*LN(RONE)) * SIN(2*PI*RTWO);
```

```
    NORM := ZONE ;
```

```
    KK := 1;
```

```
  END;
```

```
END;
```

```
procedure quicksort(var Y1: ONE_DIMENSION; Lo,Hi: integer);
```

```
procedure sort(l,r: integer);
```

```
var
```

```
  i,j : integer;
```

```
  x,yy : REAL;
```

```
begin
```

```
  i:=l; j:=r; x:=Y1[(l+r) DIV 2];
```

```
  repeat
```

```
    while Y1[i]<x do i:=i+1;
```

```
    while x<Y1[j] do j:=j-1;
```

```
    if i<=j then
```

```
      begin
```

```
        yy:=Y1[i]; Y1[i]:=Y1[j]; Y1[j]:=yy;
```

```
        i:=i+1; j:=j-1;
```

```
      end;
```

```
    until i>j;
```

```
    if l<j then sort(l,j);
```

```
    if i<r then sort(i,r);
```

```
end;
```

```
begin {quicksort};
```

```
  sort(Lo,Hi);
```

```
end;
```

```

[*****]
[**  MAIN  SUBROUTINE  **]
[**    OF SIMULATE DATA    **]
[*****]

PROCEDURE SIM_NORM;
VAR
  SUM_Y,SUM_Y2 : REAL;
  NORM : REAL;
  SUM_YDEGREE_1, SUM_YDEGREE_2 : REAL;
  MEANY , STDY2 , STDY : REAL;
  STD_Y,STD_Y2 : REAL;
BEGIN { MAIN SUBPROGRAM MAKING DATA X1,X2,Y }
  FOR J := 1 TO I_COUNT DO
  BEGIN
    NORMAL(NORM);
    X[J] := NORM;
  END;
  quicksort(X,1,I_COUNT);
  FOR J := 1 TO I_COUNT DO
  BEGIN
    NORMAL(NORM);
    YG[J] := NORM;
  END;
  ASSIGN (FILEXY,'XY1D01');
  REWRITE(FILEXY);
  FOR J := 1 TO I_COUNT DO
  BEGIN
    {** MAKING DATA X **}
    RECXY.X1 := X[J];
    {** MAKING DATA Y **}
    RECXY.Y := CORR_X_Y[K] * X[J] + (1 - SQR(CORR_X_Y[K]))*YG[J];
    WRITE(FILEXY,RECXY);
  END;
  CLOSE(FILEXY);
  TEST_CORR;
  ASSIGN(FILEXY,'XY1D01');
  RESET(FILEXY);
  SUM_YDEGREE_1 := 0;
  SUM_YDEGREE_2 := 0;
  WHILE NOT EOF(FILEXY) DO
  BEGIN
    READ(FILEXY,RECXY);
    SUM_YDEGREE_1 := SUM_YDEGREE_1 + RECXY.Y;
    SUM_YDEGREE_2 := SUM_YDEGREE_2 + SQR(RECXY.Y);
  END;
  MEANY := SUM_YDEGREE_1 / I_COUNT;
  STDY2 := (SUM_YDEGREE_2 - (I_COUNT * SQR(MEANY))) / (I_COUNT - 1);
  STDY := SQRT(STDY2);
  SUM_Y := 0.0;
  SUM_Y2 := 0.0;
  RESET(FILEXY);
  WHILE NOT EOF(FILEXY) DO
  BEGIN
    READ(FILEXY,RECXY);

```

```

RECY.Y := ((RECY.Y - MEANY) / STDY) * I_STD_NORM[K] + I_MEAN_NORM;
SUM_Y  := SUM_Y + RECY.Y;
SUM_Y2 := SUM_Y2 + SQR(RECY.Y);
SEEK(FILEXY, FILEPOS(FILEXY)-1);
WRITE(FILEXY, RECY);
END;
CLOSE(FILEXY);
MEAN_Y := SUM_Y / I_COUNT;
STD_Y  := (SUM_Y2 - (I_COUNT * SQR(MEAN_Y))) / (I_COUNT - 1);
STD_Y2 := SQR(STD_Y);
WRITELN (LST, 'MEAN POP = ', MEAN_Y:9:2, ', STD POP = ', STD_Y2:5:2);
END;

[**** TRANSFER X1 ****]

PROCEDURE TRANX (G:INTEGER);
BEGIN
  { ** 12 LEVEL ** }
  ASSIGN(FILEXY, 'XY1D01');
  RESET(FILEXY);
  CASE G OF
    1 :
      WHILE NOT EOF(FILEXY) DO
        BEGIN

          READ(FILEXY, RECY);
          WITH RECY DO
            BEGIN
              J := FILEPOS(FILEXY);
              IF X1 <= -1.38 THEN X[J] := 1.0
                ELSE IF (X1 <= -0.97) THEN X[J] := 2.0
                  ELSE IF (X1 <= -0.67) THEN X[J] := 3.0
                    ELSE IF (X1 <= -0.43) THEN X[J] := 4.0
                      ELSE IF (X1 <= -0.21) THEN X[J] := 5.0
                        ELSE IF (X1 <= 0) THEN X[J] := 6.0
                          ELSE IF (X1 <= 0.21) THEN X[J] := 7.0
                            ELSE IF (X1 <= 0.43) THEN X[J] := 8.0
                              ELSE IF (X1 <= 0.67) THEN X[J] := 9.0
                                ELSE IF (X1 <= 0.97) THEN X[J] := 10.0
                                  ELSE IF (X1 <= 1.38) THEN X[J] := 11.0
                                    ELSE X[J] := 12.0;
            END;
          END;
        { ** 9 LEVEL ** }
        2 :
          WHILE NOT EOF(FILEXY) DO
            BEGIN
              READ(FILEXY, RECY);
              WITH RECY DO
                BEGIN
                  J := FILEPOS(FILEXY);
                  IF (X1 <= -1.22) THEN X[J] := 1.0
                    ELSE IF (X1 <= -0.76) THEN X[J] := 2.0
                      ELSE IF (X1 <= -0.43) THEN X[J] := 3.0

```



```

ELSE IF (X1 <= -0.14) THEN X[J] := 4.0
ELSE IF (X1 <= 0.14) THEN X[J] := 5.0
ELSE IF (X1 <= 0.43) THEN X[J] := 6.0
ELSE IF (X1 <= 0.76) THEN X[J] := 7.0
ELSE IF (X1 <= 1.22) THEN X[J] := 8.0
ELSE X[J] := 9.0;

END;
END;
{ ** 6 LEVEL ** }
3 :
WHILE NOT EOF(FILEXY) DO
BEGIN
READ(FILEXY,RECY);
WITH RECY DO
BEGIN
J := FILEPOS(FILEXY);
IF (X1 <= -0.97) THEN X[J] := 1.0
ELSE IF (X1 <= -0.43) THEN X[J] := 2.0
ELSE IF (X1 <= 0) THEN X[J] := 3.0
ELSE IF (X1 <= 0.43) THEN X[J] := 4.0
ELSE IF (X1 <= 0.97) THEN X[J] := 5.0
ELSE X[J] := 6.0;

END;
END;
{ ** 3 LEVEL ** }
4 :
WHILE NOT EOF(FILEXY) DO
BEGIN
READ(FILEXY,RECY);
WITH RECY DO
BEGIN
J := FILEPOS(FILEXY);
IF (X1 <= -0.43) THEN X[J] := 1.0
ELSE IF (X1 <= 0.43) THEN X[J] := 2.0
ELSE X[J] := 3.0;

END;
END;
END; { END CASE }
CLOSE(FILEXY);
END;
{ ***** }

PROCEDURE RAN_STRAT(NOG,S:INTEGER);
VAR
NN          : INTEGER;
SUM_SAM     : REAL;
MEAN_STRAT1 : REAL;
ERROR_STRAT1 : REAL;

PROCEDURE RAN_SAM(NO_SAM,NOS,NN : INTEGER);

LABEL 21,22,23,24,25,26,27,28,29,30,31,32;
VAR
SAM : ONE_INT;
II,JJ : INTEGER;

```

```

BEGIN
  IF NO_SAM = 0 THEN
    EXIT;
  FOR II := 1 TO NOS DO
    SAM[II] := 0;
  II := 1;
  CASE NN OF
    1:
      BEGIN
        ASSIGN(FILEGR1, 'G1D01');
        RESET(FILEGR1);
        21:
          WHILE II <= NO_SAM DO { RANDOM SAMPLE }
            BEGIN
              REPEAT
                JJ := RANDOM(NOS+1);
              UNTIL JJ > 0;
              IF SAM[JJ] > 0 THEN
                GOTO 21;
              SEEK(FILEGR1, JJ-1);
              READ(FILEGR1, RECGR1);
              SUM_SAM := SUM_SAM + RECGR1.GROUP1;
              SAM[JJ] := 1;
              II := II + 1;
            END;
          CLOSE(FILEGR1);
        END;
      2:
        BEGIN
          ASSIGN(FILEGR2, 'G1D02');
          RESET(FILEGR2);
          22:
            WHILE II <= NO_SAM DO { RANDOM SAMPLE }
              BEGIN
                REPEAT
                  JJ := RANDOM(NOS+1);
                UNTIL JJ > 0;
                IF SAM[JJ] > 0 THEN
                  GOTO 22;
                SEEK(FILEGR2, JJ-1);
                READ(FILEGR2, RECGR2);
                SUM_SAM := SUM_SAM + RECGR2.GROUP2;
                SAM[JJ] := 1;
                II := II + 1;
              END;
            CLOSE(FILEGR2);
          END;
        3:
          BEGIN
            ASSIGN(FILEGR3, 'G1D03');
            RESET(FILEGR3);
            23:
              WHILE II <= NO_SAM DO { RANDOM SAMPLE }
                BEGIN
                  REPEAT

```

```

        JJ := RANDOM(NOS+1);
    UNTIL JJ > 0;
    IF SAM[JJ] > 0 THEN
        GOTO 23;
    SEEK(FILEGR3, JJ-1);
    READ(FILEGR3, RECGR3);
    SUM_SAM := SUM_SAM + RECGR3.GROUP3;
    SAM[JJ] := 1;
    II := II + 1;
END;
CLOSE(FILEGR3);
4:
BEGIN
    ASSIGN(FILEGR4, 'G1D04');
    RESET(FILEGR4);
24:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        REPEAT
            JJ := RANDOM(NOS+1);
        UNTIL JJ > 0;
        IF SAM[JJ] > 0 THEN
            GOTO 24;
        SEEK(FILEGR4, JJ-1);
        READ(FILEGR4, RECGR4);
        SUM_SAM := SUM_SAM + RECGR4.GROUP4;
        SAM[JJ] := 1;
        II := II + 1;
    END;
    CLOSE(FILEGR4);
END;
5:
BEGIN
    ASSIGN(FILEGR5, 'G1D05');
    RESET(FILEGR5);
25:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        REPEAT
            JJ := RANDOM(NOS+1);
        UNTIL JJ > 0;
        IF SAM[JJ] > 0 THEN
            GOTO 25;
        SEEK(FILEGR5, JJ-1);
        READ(FILEGR5, RECGR5);
        SUM_SAM := SUM_SAM + RECGR5.GROUP5;
        SAM[JJ] := 1;
        II := II + 1;
    END;
    CLOSE(FILEGR5);
END;
6:
BEGIN
    ASSIGN(FILEGR6, 'G1D06');

```

```

RESET(FILEGR6);
26:
WHILE II <= NO_SAM DO { RANDOM SAMPLE }
BEGIN
  REPEAT
    JJ := RANDOM(NOS+1);
  UNTIL JJ > 0;
  IF SAM[JJ] > 0 THEN
    GOTO 26;
  SEEK(FILEGR6, JJ-1);
  READ(FILEGR6, RECGR6);
  SUM_SAM := SUM_SAM + RECGR6.GROUP6;
  SAM[JJ] := 1;
  II := II + 1;
END;
CLOSE(FILEGR6);
END;
7:
BEGIN
  ASSIGN(FILEGR7, 'G1D07');
  RESET(FILEGR7);
27:
WHILE II <= NO_SAM DO { RANDOM SAMPLE }
BEGIN
  REPEAT
    JJ := RANDOM(NOS+1);
  UNTIL JJ > 0;
  IF SAM[JJ] > 0 THEN
    GOTO 27;
  SEEK(FILEGR7, JJ-1);
  READ(FILEGR7, RECGR7);
  SUM_SAM := SUM_SAM + RECGR7.GROUP7;
  SAM[JJ] := 1;
  II := II + 1;
END;
CLOSE(FILEGR7);
END;
8:
BEGIN
  ASSIGN(FILEGR8, 'G1D08');
  RESET(FILEGR8);
28:
WHILE II <= NO_SAM DO { RANDOM SAMPLE }
BEGIN
  REPEAT
    JJ := RANDOM(NOS+1);
  UNTIL JJ > 0;
  IF SAM[JJ] > 0 THEN
    GOTO 28;
  SEEK(FILEGR8, JJ-1);
  READ(FILEGR8, RECGR8);
  SUM_SAM := SUM_SAM + RECGR8.GROUP8;
  SAM[JJ] := 1;
  II := II + 1;
END;

```

```

        CLOSE(FILEGR8);
    END;
    9:
    BEGIN
        ASSIGN(FILEGR9,'G1D09');
        RESET(FILEGR9);
        29:
        WHILE II <= NO_SAM DO { RANDOM SAMPLE }
        BEGIN
            REPEAT
                JJ := RANDOM(NOS+1);
            UNTIL JJ > 0;
            IF SAM[JJ] > 0 THEN
                GOTO 29;
            SEEK(FILEGR9,JJ-1);
            READ(FILEGR9,RECGR9);
            SUM_SAM := SUM_SAM + RECGR9.GROUP9;
            SAM[JJ] := 1;
            II := II + 1;
        END;
        CLOSE(FILEGR9);
    END;
    10:
    BEGIN
        ASSIGN(FILEGR10,'G1D010');
        RESET(FILEGR10);
        30:
        WHILE II <= NO_SAM DO { RANDOM SAMPLE }
        BEGIN
            REPEAT
                JJ := RANDOM(NOS+1);
            UNTIL JJ > 0;
            IF SAM[JJ] > 0 THEN
                GOTO 30;
            SEEK(FILEGR10,JJ-1);
            READ(FILEGR10,RECGR10);
            SUM_SAM := SUM_SAM + RECGR10.GROUP10;
            SAM[JJ] := 1;
            II := II + 1;
        END;
        CLOSE(FILEGR10);
    END;
    11:
    BEGIN
        ASSIGN(FILEGR11,'G1D011');
        RESET(FILEGR11);
        31:
        WHILE II <= NO_SAM DO { RANDOM SAMPLE }
        BEGIN
            REPEAT
                JJ := RANDOM(NOS+1);
            UNTIL JJ > 0;
            IF SAM[JJ] > 0 THEN
                GOTO 31;
            SEEK(FILEGR11,JJ-1);

```

```

        READ(FILEGR11,RECGR11);
        SUM_SAM := SUM_SAM + RECGR11.GROUP11;
        SAM[JJ] := 1;
        II := II + 1;
    END;
    CLOSE(FILEGR11);
END;
12:
BEGIN
    ASSIGN(FILEGR12,'G1D012');
    RESET(FILEGR12);
32:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        REPEAT
            JJ := RANDOM(NOS+1);
        UNTIL JJ > 0;
        IF SAM[JJ] > 0 THEN
            GOTO 32;
        SEEK(FILEGR12, JJ-1);
        READ(FILEGR12,RECGR12);
        SUM_SAM := SUM_SAM + RECGR12.GROUP12;
        SAM[JJ] := 1;
        II := II + 1;
    END;
    CLOSE(FILEGR12);
END;
    END; { END CASE }
END;
{*****END RAN_SAM*****}

BEGIN
    SUM_SAM := 0.0;
    SUM_STDST := 0.0;
    FOR NN := 1 TO NOG DO
        RAN_SAM(NO_SAM[NN],NO[NN],NN);
    MEAN_STRAT1 := SUM_SAM / I_SAM[I];
    ERROR_STRAT1 := ABS(MEAN_STRAT1 - I_MEAN_NORM) * 100 / I_MEAN_NORM;
    MEAN_STRAT := MEAN_STRAT + MEAN_STRAT1;
    ERROR_STRAT := ERROR_STRAT + ERROR_STRAT1;
END;
{*****END RAN_STRAT*****}

PROCEDURE RAN_QUOTA(NOG,S:INTEGER);
VAR
    SS                               : INTEGER;
    MEAN_QUOTA1,SUM_QSAM              : REAL;
    ERROR_QUOTA1                     : REAL;
    STD_QUOTA1                       : REAL;
    NN                               : INTEGER;
    STD_QT,MEAN_QT,SUM_QT,SUM_QT2    : REAL;

```

```
PROCEDURE QUOTA_SAM(NO_SAM,NOS,NN : INTEGER);
```

```
  LABEL 41,42,43,44,45,46,47,48,49,50,51,52;
```

```
  VAR
```

```
    II,MID : INTEGER;
```

```
    ADDJ,SUBJ,MAX,MIN,FLAG : INTEGER;
```

```
  BEGIN
```

```
    IF NO_SAM = 0 THEN
```

```
      EXIT;
```

```
    MIN := 1;
```

```
    MAX := NOS;
```

```
    II := 1;
```

```
    FLAG := 1;
```

```
    CASE NN OF
```

```
      1:
```

```
        BEGIN
```

```
          ASSIGN(FILEGR1,'G1D01');
```

```
          RESET(FILEGR1);
```

```
          41:
```

```
          WHILE II <= NO_SAM DO { RANDOM SAMPLE }
```

```
          BEGIN
```

```
            CASE FLAG OF
```

```
              1 :
```

```
                BEGIN
```

```
                  REPEAT
```

```
                    MID := RANDOM(MAX + 1);
```

```
                  UNTIL MID >= MIN;
```

```
                  IF MID = MAX THEN
```

```
                    MAX := MAX - 1
```

```
                  ELSE
```

```
                  IF MID = MIN THEN
```

```
                    MIN := MIN + 1
```

```
                  ELSE
```

```
                    BEGIN
```

```
                      ADDJ := MID;
```

```
                      SUBJ := MID;
```

```
                      FLAG := 2;
```

```
                    END;
```

```
                END;
```

```
              2 :
```

```
                BEGIN
```

```
                  ADDJ := ADDJ + 1;
```

```
                  IF ADDJ > MAX THEN
```

```
                    BEGIN
```

```
                      MAX := SUBJ-1;
```

```
                      FLAG := 1;
```

```
                      GOTO 41;
```

```
                    END
```

```
                  ELSE
```

```
                    BEGIN
```

```
                      FLAG := 3;
```

```
                      MID := ADDJ;
```

```
                    END;
```

```
                END;
```

```
            END;
```

```
          END;
```

```

3 :
BEGIN
  SUBJ := SUBJ - 1;
  IF SUBJ = MIN THEN
  BEGIN
    FLAG := 1;
    MIN := ADDJ+1;
  END
  ELSE
    FLAG := 2;
    MID := SUBJ;
  END;
END; { END CASE }
SEEK(FILEGR1,MID-1);
READ(FILEGR1,RECGR1);
SUM_QSAM := SUM_QSAM + RECGR1.GROUP1;
II := II + 1;
END;
CLOSE(FILEGR1);
END;
2:
BEGIN
  ASSIGN(FILEGR2,'G1D02');
  RESET(FILEGR2);
  42:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    CASE FLAG OF
      1 :
      BEGIN
        REPEAT
          MID := RANDOM(MAX + 1);
        UNTIL MID >= MIN;
        IF MID = MAX THEN
          MAX := MAX - 1;
        ELSE
          IF MID = MIN THEN
            MIN := MIN + 1;
          ELSE
            BEGIN
              ADDJ := MID;
              SUBJ := MID;
              FLAG := 2;
            END;
          END;
        END;
      2 :
      BEGIN
        ADDJ := ADDJ + 1;
        IF ADDJ > MAX THEN
          BEGIN
            MAX := SUBJ-1;
            FLAG := 1;
            GOTO 42;
          END
        ELSE

```



```

        BEGIN
            FLAG := 3;
            MID := ADDJ;
        END;
    END;
3 :
    BEGIN
        SUBJ := SUBJ - 1;
        IF SUBJ = MIN THEN
            BEGIN
                FLAG := 1;
                MIN := ADDJ+1;
            END
        ELSE
            FLAG := 2;
            MID := SUBJ;
        END;
    END; { END CASE }
    SEEK(FILEGR2,MID-1);
    READ(FILEGR2,RECGR2);
    SUM_QSAM := SUM_QSAM + RECGR2.GROUP2;
    II := II + 1;
END;
CLOSE(FILEGR2);
END;
3:
BEGIN
    ASSIGN(FILEGR3,'G1D03');
    RESET(FILEGR3);
    43:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
                BEGIN
                    REPEAT
                        MID := RANDOM(MAX + 1);
                    UNTIL MID >= MIN;
                    IF MID = MAX THEN
                        MAX := MAX - 1
                    ELSE
                        IF MID = MIN THEN
                            MIN := MIN + 1
                        ELSE
                            BEGIN
                                ADDJ := MID;
                                SUBJ := MID;
                                FLAG := 2;
                            END;
                        END;
                    END;
                END;
            2 :
                BEGIN
                    ADDJ := ADDJ + 1;
                    IF ADDJ > MAX THEN
                        BEGIN

```

```

        MAX := SUBJ-1;
        FLAG := 1;
        GOTO 43;
    END
    ELSE
    BEGIN
        FLAG := 3;
        MID := ADDJ;
    END;
END;
3 :
BEGIN
    SUBJ := SUBJ - 1;
    IF SUBJ = MIN THEN
    BEGIN
        FLAG := 1;
        MIN := ADDJ+1;
    END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR3,MID-1);
READ(FILEGR3,RECGR3);
SUM_QSAM := SUM_QSAM + RECGR3.GROUP3;
II := II + 1;
END;
CLOSE(FILEGR3);
END;
4:
BEGIN
    ASSIGN(FILEGR4,'G1D04');
    RESET(FILEGR4);
    44:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
            BEGIN
                REPEAT
                    MID := RANDOM(MAX + 1);
                UNTIL MID >= MIN;
                IF MID = MAX THEN
                    MAX := MAX - 1
                ELSE
                    IF MID = MIN THEN
                        MIN := MIN + 1
                    ELSE
                        BEGIN
                            ADDJ := MID;
                            SUBJ := MID;
                            FLAG := 2;
                        END;
                    END;
                END;
            END;
        END;
    END;
END;

```

```

2 :
BEGIN
  ADDJ := ADDJ + 1;
  IF ADDJ > MAX THEN
  BEGIN
    MAX := SUBJ-1;
    FLAG := 1;
    GOTO 44;
  END
  ELSE
  BEGIN
    FLAG := 3;
    MID := ADDJ;
  END;
END;
3 :
BEGIN
  SUBJ := SUBJ - 1;
  IF SUBJ = MIN THEN
  BEGIN
    FLAG := 1;
    MIN := ADDJ+1;
  END
  ELSE
    FLAG := 2;
    MID := SUBJ;
  END;
END; { END CASE }
SEEK(FILEGR4,MID-1);
READ(FILEGR4,RECGR4);
SUM_QSAM := SUM_QSAM + RECGR4.GROUP4;
II := II + 1;
END;
CLOSE(FILEGR4);
END;
5:
BEGIN
  ASSIGN(FILEGR5,'G1D05');
  RESET(FILEGR5);
  45:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    CASE FLAG OF
      1 :
      BEGIN
        REPEAT
          MID := RANDOM(MAX + 1);
        UNTIL MID >= MIN;
        IF MID = MAX THEN
          MAX := MAX - 1
        ELSE
          IF MID = MIN THEN
            MIN := MIN + 1
          ELSE
            BEGIN

```

```

        ADDJ := MID;
        SUBJ := MID;
        FLAG := 2;
    END;
END;
2 :
BEGIN
    ADDJ := ADDJ + 1;
    IF ADDJ > MAX THEN
    BEGIN
        MAX := SUBJ-1;
        FLAG := 1;
        GOTO 45;
    END
    ELSE
    BEGIN
        FLAG := 3;
        MID := ADDJ;
    END;
END;
3 :
BEGIN
    SUBJ := SUBJ - 1;
    IF SUBJ = MIN THEN
    BEGIN
        FLAG := 1;
        MIN := ADDJ+1;
    END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR5,MID-1);
READ(FILEGR5,RECGR5);
SUM_QSAM := SUM_QSAM + RECGR5.GROUP5;
II := II + 1;
END;
CLOSE(FILEGR5);
END;
6:
BEGIN
    ASSIGN(FILEGR6,'G1D06');
    RESET(FILEGR6);
46:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
                BEGIN
                    REPEAT
                        MID := RANDOM(MAX + 1);
                    UNTIL MID >= MIN;
                    IF MID = MAX THEN
                        MAX := MAX - 1

```

```

ELSE
  IF MID = MIN THEN
    MIN := MIN + 1
  ELSE
    BEGIN
      ADDJ := MID;
      SUBJ := MID;
      FLAG := 2;
    END;
  END;
2 :
BEGIN
  ADDJ := ADDJ + 1;
  IF ADDJ > MAX THEN
    BEGIN
      MAX := SUBJ-1;
      FLAG := 1;
      GOTO 46;
    END
  ELSE
    BEGIN
      FLAG := 3;
      MID := ADDJ;
    END;
  END;
3 :
BEGIN
  SUBJ := SUBJ - 1;
  IF SUBJ = MIN THEN
    BEGIN
      FLAG := 1;
      MIN := ADDJ+1;
    END
  ELSE
    BEGIN
      FLAG := 2;
      MID := SUBJ;
    END;
  END;
END; { END CASE }
SEEK(FILEGR6,MID-1);
READ(FILEGR6,RECGR6);
SUM_QSAM := SUM_QSAM + RECGR6.GROUP6;
II := II + 1;
END;
CLOSE(FILEGR6);
END;
7:
BEGIN
  ASSIGN(FILEGR7,'G1D07');
  RESET(FILEGR7);
47:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    CASE FLAG OF
      1 :
        BEGIN

```

```

[*****]
[**          THESIS          **]
[** COMPARATIVE STRTITY WITH QUOTA SAMPLING **]
[**      BY MR. PRACHA SUWATTANAPUNGUL      **]
[*****]

PROGRAM STRATIFY_QUOTA(INPUT,OUTPUT);  [** FACTOR2 **]
USES  CRT,PRINTER;
CONST
  I_COUNT      = 5000; {TOTAL OBSERVATION IN EACH VARIABLE (N)}
  K_COUNT      = 12;  {TOTAL POPULATION}
  S_COUNT      = 25;  {REPEAT RANDOM STRAT}
  I_MEAN_NORM  = 10000.00; {MEAN OF DATA Y}
  I_TOT_GROUP_X1 = 4;  {TOTAL GROUP OF X1}
  I_TOT_GROUP_X2 = 3;  {TOTAL GROUP OF X2}

TYPE
  VARXY      = RECORD
                X1      : REAL;
                X2      : REAL;
                Y       : REAL;
            END;
  VARGR1     = RECORD
                GROUP1 : REAL;
            END;
  VARGR2     = RECORD
                GROUP2 : REAL;
            END;
  VARGR3     = RECORD
                GROUP3 : REAL;
            END;
  VARGR4     = RECORD
                GROUP4 : REAL;
            END;
  VARGR5     = RECORD
                GROUP5 : REAL;
            END;
  VARGR6     = RECORD
                GROUP6 : REAL;
            END;
  VARGR7     = RECORD
                GROUP7 : REAL;
            END;
  VARGR8     = RECORD
                GROUP8 : REAL;
            END;
  VARGR9     = RECORD
                GROUP9 : REAL;
            END;
  VARGR10    = RECORD
                GROUP10: REAL;
            END;
  VARGR11    = RECORD

```

```

                                GROUP11: REAL;
END;
VARGR12      = RECORD
                                GROUP12: REAL;
END;
ONE_DIMENSION = ARRAY [1..I_COUNT] OF REAL;
ONE_INT       = ARRAY [1..I_COUNT] OF BYTE;
ONE_D2        = ARRAY [1..12] OF INTEGER;
ONE_D3        = ARRAY [1..3] OF REAL;
ONE_D4        = ARRAY [1..6] OF REAL;
ONE_STD       = ARRAY [1..12] OF REAL;
ST4           = STRING[4];
VAR
FILEXY       : FILE OF VARXY;
RECY        : VARXY;
FILEGR1      : FILE OF VARGR1;
RECGR1       : VARGR1;
FILEGR2      : FILE OF VARGR2;
RECGR2       : VARGR2;
FILEGR3      : FILE OF VARGR3;
RECGR3       : VARGR3;
FILEGR4      : FILE OF VARGR4;
RECGR4       : VARGR4;
FILEGR5      : FILE OF VARGR5;
RECGR5       : VARGR5;
FILEGR6      : FILE OF VARGR6;
RECGR6       : VARGR6;
FILEGR7      : FILE OF VARGR7;
RECGR7       : VARGR7;
FILEGR8      : FILE OF VARGR8;
RECGR8       : VARGR8;
FILEGR9      : FILE OF VARGR9;
RECGR9       : VARGR9;
FILEGR10     : FILE OF VARGR10;
RECGR10      : VARGR10;
FILEGR11     : FILE OF VARGR11;
RECGR11      : VARGR11;
FILEGR12     : FILE OF VARGR12;
RECGR12      : VARGR12;
I,J,K,L,M,N,KK: INTEGER;
G : INTEGER;
NO_SAM       : ONE_D2;
NO,NOQ      : ONE_D2;
{ X1,X2      : ONE_DIMENSION; }
YG          : ONE_DIMENSION;
IX,IX1,IX2  : LONGINT;
STD_Y,STD_Y2 : REAL;
MEAN_Y,MEAN_SAM : REAL;
{ R_MEAN,SUMR      : ONE_D3;
SUM_ERROR_MEANS,SUM_ERROR_QMEANS : ONE_D4;
SUM_MEANS,SUM_QMEANS      : ONE_D4;
S_MEAN,S_ERROR_MEAN      : ONE_D4;
Q_MEAN,Q_ERROR_MEAN      : ONE_D4; }
MEAN_STRAT,MEAN_QUOTA : REAL;
ERROR_STRAT,ERROR_QUOTA : REAL;

```

```

CORR_X1_X2 : ONE_STD; {DEGREE CORRELATION COEFFICIENT BETWEEN X1 AND X2}
CORR_X1_Y  : ONE_STD; {DEGREE CORRELATION COEFFICIENT BETWEEN Y AND X1}
CORR_X2_Y  : ONE_STD; {DEGREE CORRELATION COEFFICIENT BETWEEN Y AND X2}
{**** C.V. = 2%,5%,10%,20% ****}
I_STD_NORM : ONE_STD; {STANDART DIVATION OF DATA Y = 200,500,1000,2000}
I_SAM      : ONE_D2; {SAMPLE n = 250,500,750 }
CVP : REAL;

```

```

{*****}
{**      ROUTINE SIMULATE 4      **}
{**  N O R M A L    D I S T R I B U T I O N  **}
{**      F O R      **}
{**  I N D E P E N D E N T  V A R I A B L E  ---> X1,X2  **}
{**  D E P E N D E N T  V A R I A B L E  ---> Y      **}
{*****}

```

```

{*****}
{**      SUBROUTINE FOR MAKING      **}
{**  R A N D O M    N U M B E R      **}
{*****}

```

```

FUNCTION RAND(IXX : LONGINT) : REAL;
BEGIN

```

```

    IX := IXX * 16807;
    IF (IX < 0) THEN
        IX := IX + 2147483647 + 1;
    RAND := IX * 0.4656613E-9;
    |
    RAND := IX / 2147483647;|

```

```

END;
FUNCTION RANDS(IXS : LONGINT) : REAL;
BEGIN

```

```

    IX1 := IXS * 16807;
    IF (IX1 < 0) THEN
        IX1 := IX1 + 2147483647 + 1;
    RANDS := IX1 * 0.4656613E-9;

```

```

END;
FUNCTION RANDQ(IXQ : LONGINT) : REAL;
BEGIN

```

```

    IX2 := IXQ * 16807;
    IF (IX2 < 0) THEN
        IX2 := IX2 + 2147483647 + 1;
    RANDQ := IX2 * 0.4656613E-9;

```

```

END;

```

```

{*****}
{**      SUBROUTINE FOR MAKING      **}
{**  N O R M A L    ( 0 , 1 )      **}
{*****}

```

```

PROCEDURE NORMAL(VAR NORM : REAL);

```

```

CONST PI = 3.1415926;

```

```

VAR

```

```

    RONE,RTWO : REAL;
    ZONE,ZTWO : REAL;
    RAN_NUM   : REAL;

```



```

BEGIN
[   IF KK = 1 THEN]
[   BEGIN]
[       NORM := ZTWO ;]
[       KK := 0;]
[   END]
[   ELSE]
[   BEGIN]
[       RONE := RAND(IX);
[       RTWO := RAND(IX);
[       ZONE := SQRT(-2*LN(RONE)) * COS(2*PI*RTWO);
[       ZTWO := SQRT(-2*LN(RONE)) * SIN(2*PI*RTWO);]
[       NORM := ZONE ;
[       KK := 1;]
[   END;]
END;
[*****]
procedure quicksort(var Y1: ONE_DIMENSION; Lo,Hi: integer);

procedure sort(l,r: integer);
var
    i,j : integer;
    x,yy : REAL;
begin
    i:=l; j:=r; x:=Y1[(l+r) DIV 2];
    repeat
        while Y1[i]<x do i:=i+1;
        while x<Y1[j] do j:=j-1;
        if i<=j then
            begin
                yy:=Y1[i]; Y1[i]:=Y1[j]; Y1[j]:=yy;
                i:=i+1; j:=j-1;
            end;
        until i>j;
        if l<j then sort(l,j);
        if i<r then sort(i,r);
    end;

begin {quicksort};
    sort(Lo,Hi);
end;
[*****]

[*****]
[** SUBPROGRAM TEST MULTICOLINEARITY **]
[**   OF SIMULATE DATA           **]
[*****]
PROCEDURE TEST_CORR(I : INTEGER; NAME : ST4;CORR1 : REAL);
VAR
    SUMXY,SUMX,SUMY,SUMXX,SUMYY : REAL ;
    CORR :REAL;
    NAME_R : ST4;
    R :REAL;
BEGIN
    R := 0.0;

```

```

SUMXY := 0;
SUMX  := 0;
SUMY  := 0;
SUMXX := 0;
SUMYY := 0;
ASSIGN(FILEXY, 'XY1D01');
RESET(FILEXY);
IF NAME = 'R_12' THEN
  WHILE NOT EOF(FILEXY) DO
    BEGIN
      READ(FILEXY, RECY);
      SUMXY := SUMXY + RECY.X1 * RECY.X2;
      SUMXX := SUMXX + RECY.X1 * RECY.X1;
      SUMYY := SUMYY + RECY.X2 * RECY.X2;
      SUMY  := SUMY  + RECY.X2;
      SUMX  := SUMX  + RECY.X1;
    END
  ELSE
    IF NAME = 'R_Y1' THEN
      WHILE NOT EOF(FILEXY) DO
        BEGIN
          READ(FILEXY, RECY);
          SUMXY := SUMXY + RECY.Y * RECY.X1;
          SUMXX := SUMXX + RECY.Y * RECY.Y;
          SUMYY := SUMYY + RECY.X1 * RECY.X1;
          SUMY  := SUMY  + RECY.X1;
          SUMX  := SUMX  + RECY.Y;
        END
      ELSE
        WHILE NOT EOF(FILEXY) DO
          BEGIN
            READ(FILEXY, RECY);
            SUMXY := SUMXY + RECY.Y * RECY.X2;
            SUMXX := SUMXX + RECY.Y * RECY.Y;
            SUMYY := SUMYY + RECY.X2 * RECY.X2;
            SUMY  := SUMY  + RECY.X2;
            SUMX  := SUMX  + RECY.Y;
          END;
        CLOSE(FILEXY);
        R := I_COUNT * SUMXY - SUMX * SUMY;
        R := R/SQRT((I_COUNT*SUMXX - SQR(SUMX))*(I_COUNT*SUMYY - SQR(SUMY)));
        { SUMR[I] := SUMR[I] + R;}
        NAME_R := NAME;
        CORR   := CORR1;
        WRITELN('VALUE CORRRELATION SETTING = ', CORR:5:3);
        WRITELN(NAME_R, ' = ', R:5:3);
        WRITELN(LST, 'VALUE CORRRELATION SETTING = ', CORR:5:3);
        WRITELN(LST, NAME_R, ' = ', R:5:3);
      END;
    {*****}

```

```

{*****}
{**  MAIN  SUBROUTINE  **}
{**    OF SIMULATE DATA    **}
{*****}

PROCEDURE SIM_NORM;

TYPE {VARZ1 = RECORD
      Z1 : REAL;
    END;}
    VARZ2 = RECORD
      Z2 : REAL;
    END;
    VARZ3 = RECORD
      Z3 : REAL;
    END;

VAR
{
  FILEZ1 : FILE OF VARZ1;
  RECZ1  : VARZ1;
  FILEZ2 : FILE OF VARZ2;
  RECZ2  : VARZ2;
  FILEZ3 : FILE OF VARZ3;
  RECZ3  : VARZ3;
  NORM,SUMX2,SUMY  : REAL;
  SUM_Y,SUM_Y2  : REAL;
  TMP  : REAL;
  TMP1 :REAL;
  TMP2 : REAL;
  TMP3 : REAL;
  TMP4 : REAL;
  TMP5 : REAL;
  SUM_X2DEGREE_1, SUM_X2DEGREE_2  : REAL;
  MEANX2 , STDY2 , STDY  : REAL;
  SUM_YDEGREE_1, SUM_YDEGREE_2  : REAL;
  MEANY , STDY2 , STDY  : REAL;

BEGIN { MAIN SUBPROGRAM MAKING DATA X1,X2,Y) }

{*** SIMULATE Z1,Z2,Z3 ~ N(0,1) ***}
{  ASSIGN (FILEZ1,'Z1D01');
  REWRITE(FILEZ1);}
  FOR J := 1 TO I_COUNT DO
  BEGIN
    NORMAL(NORM);
  {
    RECZ1.Z1 := NORM;
    WRITE(FILEZ1,RECZ1);}
    YG[J] :='NORM;
  END;
  {  CLOSE (FILEZ1);}
  quicksort(yg,i,i_count);
  ASSIGN (FILEZ2,'Z1D02');
  REWRITE(FILEZ2);
  FOR J := 1 TO I_COUNT DO
  BEGIN
    NORMAL(NORM);

```

```

SUM_YDEGREE_1 := 0;
SUM_YDEGREE_2 := 0;
WHILE NOT EOF(FILEXY) DO
BEGIN
  READ(FILEXY,RECY);
  SUM_X2DEGREE_1 := SUM_X2DEGREE_1 + RECY.X2;
  SUM_X2DEGREE_2 := SUM_X2DEGREE_2 + SQR(RECY.X2);
  SUM_YDEGREE_1 := SUM_YDEGREE_1 + RECY.Y;
  SUM_YDEGREE_2 := SUM_YDEGREE_2 + SQR(RECY.Y);
END;
MEANX2 := SUM_X2DEGREE_1 / I_COUNT;
STDY2 := (SUM_X2DEGREE_2 - (I_COUNT * SQR(MEANX2))) / (I_COUNT - 1);
STDY2 := SQR(STDY2);
MEANY := SUM_YDEGREE_1 / I_COUNT;
STDY := (SUM_YDEGREE_2 - (I_COUNT * SQR(MEANY))) / (I_COUNT - 1);
STDY := SQR(STDY);
SUM_Y := 0.0;
SUM_Y2 := 0.0;
RESET(FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
  READ(FILEXY,RECY);
  RECY.X2 := (RECY.X2 - MEANX2) / STDY2;
  RECY.Y := ((RECY.Y - MEANY) / STDY) * I_STD_NORM[K] + I_MEAN_NORM;
  SUM_Y := SUM_Y + RECY.Y;
  SUM_Y2 := SUM_Y2 + SQR(RECY.Y);}
  SEEK(FILEXY,FILEPOS(FILEXY)-1);
  WRITE(FILEXY,RECY);
END;
CLOSE(FILEXY);
{ MEAN_Y := SUM_Y / I_COUNT;
  STD_Y := (SUM_Y2 - (I_COUNT * SQR(MEAN_Y))) / (I_COUNT - 1);
  STD_Y2 := SQR(STD_Y);
  WRITELN(LST,'MEAN POP = ',MEAN_Y:9:2,', STD POP = ',STD_Y2:5:2);}
END;

{**** TRANSFER X1 = 1,2,3,4 *** X2 = 1,2,3 ****}
PROCEDURE TRANX;
BEGIN
  ASSIGN(FILEXY,'XY1D01');
  RESET(FILEXY);
  WHILE NOT EOF(FILEXY) DO
  BEGIN
    READ(FILEXY,RECY);
    WITH RECY DO
    BEGIN
      IF X1 <= -0.67 THEN
        X1 := 1.0
      ELSE
        IF (X1 <= 0) THEN
          X1 := 2.0
        ELSE
          IF (X1 <= 0.67) THEN
            X1 := 3.0
          ELSE

```

```

                X1 := 4.0;
IF X2 <= -0.43 THEN
    X2 := 1.0
ELSE
    IF (X2 <= 0.43) THEN
        X2 := 2.0
    ELSE
        X2 := 3.0;
END;
SEEK(FILEXY, FILEPOS(FILEXY)-1);
WRITE(FILEXY, RECX);
END;
CLOSE(FILEXY);
END;
{*****}

[*** SUBPROGRAM WRITE Y X1 X2 ***]

{PROCEDURE PRTDATA;
BEGIN
    WRITELN('Y DATA ' :40);
    FOR J := 1 TO I_COUNT DO
        WRITE(' ', Y[J]:6:2);
    WRITELN;
    WRITELN('X1 DATA ' :40);
    FOR J := 1 TO I_COUNT DO
        WRITE(X1[J]:6:2);
    WRITELN;
    WRITELN('X2 DATA ' :40);
    FOR J := 1 TO I_COUNT DO
        WRITE(X2[J]:6:2);
    WRITELN;
END;}

{PROCEDURE PRTG(G:INTEGER);
CONST
    COL_NO1 : INTEGER = 0;
BEGIN
    FOR I := 1 TO G DO
        BEGIN
            WRITELN('GROUP ', I, ' = ' :40);
            FOR J := 1 TO NO[I] DO
                WRITE(1st, ' ', GROUP[I,J]:8:2);
            WRITELN;
        END;
END;}

FUNCTION QTRAN12:INTEGER;
    { X1 = 4 LEVEL , X2 = 3 LEVEL }
BEGIN
    WITH RECX DO
        BEGIN
            IF (X1=1.0) AND (X2=1.0) THEN
                QTRAN12 := 1

```

```

ELSE
IF (X1=1.0) AND (X2=2.0) THEN
  QTRAN12 := 2
ELSE
IF (X1=1.0) AND (X2=3.0) THEN
  QTRAN12 := 3
ELSE
IF (X1=2.0) AND (X2=1.0) THEN
  QTRAN12 := 4
ELSE
IF (X1=2.0) AND (X2=2.0) THEN
  QTRAN12 := 5
ELSE
IF (X1=2.0) AND (X2=3.0) THEN
  QTRAN12 := 6
ELSE
IF (X1=3.0) AND (X2=1.0) THEN
  QTRAN12 := 7
ELSE
IF (X1=3.0) AND (X2=2.0) THEN
  QTRAN12 := 8
ELSE
IF (X1=3.0) AND (X2=3.0) THEN
  QTRAN12 := 9
ELSE
IF (X1=4.0) AND (X2=1.0) THEN
  QTRAN12 := 10
ELSE
IF (X1=4.0) AND (X2=2.0) THEN
  QTRAN12 := 11
ELSE
  QTRAN12 := 12;
END;

END;
FUNCTION QTRAN7:INTEGER;
BEGIN
  WITH RECY DO
  BEGIN
  IF (X1=1.0) AND (X2=1.0) THEN
    QTRAN7 := 1
  ELSE
  IF ((X1=1.0) AND (X2=2.0)) OR ((X1=2.0) AND (X2=1.0)) THEN
    QTRAN7 := 2
  ELSE
  IF ((X1=1.0) AND (X2=3.0)) OR ((X1=3.0) AND (X2=1.0)) THEN
    QTRAN7 := 3
  ELSE
  IF ((X1=2.0) AND (X2=2.0)) OR ((X1=4.0) AND (X2=1.0)) THEN
    QTRAN7 := 4
  ELSE
  IF ((X1=2.0) AND (X2=3.0)) OR ((X1=3.0) AND (X2=2.0)) THEN
    QTRAN7 := 5
  ELSE
  IF ((X1=4.0) AND (X2=2.0)) OR ((X1=3.0) AND (X2=3.0)) THEN
    QTRAN7 := 6
  
```

```

ELSE
  IF (X1=4.0) AND (X2=3.0) THEN
    QTRAN7 := 7;
  END;
END;
FUNCTION QTRAN3:INTEGER;
BEGIN
  WITH RECY DO
  BEGIN
    IF ((X1=1.0) AND ((X2=1.0) OR (X2=2.0) OR (X2=3.0)))
      OR ((X1=2.0) AND (X2=1.0))
      OR ((X1=3.0) AND (X2=1.0)) THEN
      QTRAN3 := 1
    ELSE
      IF ((X1=2.0) AND ((X2=2.0) OR (X2=3.0)))
        OR ((X1=3.0) AND (X2=2.0))
        OR ((X1=4.0) AND (X2=1.0)) THEN
        QTRAN3 := 2
      ELSE
        QTRAN3 := 3;
      END;
    END;
  END;
  [*****]

```

```

PROCEDURE RAN_STRAT(NOG,S:INTEGER);
VAR

```

```

  NN : INTEGER;
  SUM_SAM : REAL;
  MEAN_STRAT1 : REAL;
  ERROR_STRAT1 : REAL;

```

```

PROCEDURE RAN_SAM(NO_SAM,NOS,N : INTEGER);

```

```

LABEL 21,22,23,24,25,26,27,28,29,30,31,32;

```

```

VAR

```

```

  SAM : ONE_INT;
  II,JJ : INTEGER;

```

```

BEGIN

```

```

  IF NO_SAM = 0 THEN

```

```

    EXIT;

```

```

  FOR II := 1 TO NOS DO

```

```

    SAM[II] := 0;

```

```

  II := 1;

```

```

  CASE N OF

```

```

    1:

```

```

    BEGIN

```

```

      ASSIGN(FILEGR1,'G1D01');

```

```

      RESET(FILEGR1);

```

```

      21:

```

```

      WHILE II <= NO_SAM DO { RANDOM SAMPLE }

```

```

      BEGIN

```

```

        REPEAT

```

```

          JJ := ROUND(RANDS(IX1) * NOS) ;|

```

```

        JJ := RANDOM(NOS+1);
    UNTIL JJ > 0;
    IF SAM[JJ] > 0 THEN
        GOTO 21;
    SEEK(FILEGR1, JJ-1);
    READ(FILEGR1, RECGR1);
    SUM_SAM := SUM_SAM + RECGR1.GROUP1;
    SAM[JJ] := 1;
    II := II + 1;
END;
CLOSE(FILEGR1);
END;
2:
BEGIN
    ASSIGN(FILEGR2, 'G1D02');
    RESET(FILEGR2);
    22:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        REPEAT
            JJ := ROUND(RANDS(IX1) * NOS) ;}
            JJ := RANDOM(NOS+1);
        UNTIL JJ > 0;
        IF SAM[JJ] > 0 THEN
            GOTO 22;
        SEEK(FILEGR2, JJ-1);
        READ(FILEGR2, RECGR2);
        SUM_SAM := SUM_SAM + RECGR2.GROUP2;
        SAM[JJ] := 1;
        II := II + 1;
    END;
    CLOSE(FILEGR2);
END;
3:
BEGIN
    ASSIGN(FILEGR3, 'G1D03');
    RESET(FILEGR3);
    23:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        REPEAT
            JJ := ROUND(RANDS(IX1) * NOS) ;}
            JJ := RANDOM(NOS+1);
        UNTIL JJ > 0;
        IF SAM[JJ] > 0 THEN
            GOTO 23;
        SEEK(FILEGR3, JJ-1);
        READ(FILEGR3, RECGR3);
        SUM_SAM := SUM_SAM + RECGR3.GROUP3;
        SAM[JJ] := 1;
        II := II + 1;
    END;
    CLOSE(FILEGR3);
END;
4:

```



```

BEGIN
  ASSIGN(FILEGR4,'G1D04');
  RESET(FILEGR4);
24:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    REPEAT
      JJ := ROUND(RANDS(IX1) * NOS) ;
      JJ := RANDOM(NOS+1);}
    UNTIL JJ > 0;
    IF SAM[JJ] > 0 THEN
      GOTO 24;
    SEEK(FILEGR4,JJ-1);
    READ(FILEGR4,RECGR4);
    SUM_SAM := SUM_SAM + RECGR4.GROUP4;
    SAM[JJ] := 1;
    II := II + 1;
  END;
  CLOSE(FILEGR4);
END;
5:
BEGIN
  ASSIGN(FILEGR5,'G1D05');
  RESET(FILEGR5);
25:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    REPEAT
      JJ := ROUND(RANDS(IX1) * NOS) ;}
      JJ := RANDOM(NOS+1);
    UNTIL JJ > 0;
    IF SAM[JJ] > 0 THEN
      GOTO 25;
    SEEK(FILEGR5,JJ-1);
    READ(FILEGR5,RECGR5);
    SUM_SAM := SUM_SAM + RECGR5.GROUP5;
    SAM[JJ] := 1;
    II := II + 1;
  END;
  CLOSE(FILEGR5);
END;
6:
BEGIN
  ASSIGN(FILEGR6,'G1D06');
  RESET(FILEGR6);
26:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    REPEAT
      JJ := ROUND(RANDS(IX1) * NOS) ;}
      JJ := RANDOM(NOS+1);
    UNTIL JJ > 0;
    IF SAM[JJ] > 0 THEN
      GOTO 26;
    SEEK(FILEGR6,JJ-1);

```

```

        READ(FILEGR6,RECGR6);
        SUM_SAM := SUM_SAM + RECGR6.GROUP6;
        SAM[JJ] := 1;
        II := II + 1;
    END;
    CLOSE(FILEGR6);
END;
7:
BEGIN
    ASSIGN(FILEGR7,'G1D07');
    RESET(FILEGR7);
    27:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        REPEAT
            JJ := ROUND(RANDS(IX1) * NOS) ;|
            JJ := RANDOM(NOS+1);
        UNTIL JJ > 0;
        IF SAM[JJ] > 0 THEN
            GOTO 27;
        SEEK(FILEGR7,JJ-1);
        READ(FILEGR7,RECGR7);
        SUM_SAM := SUM_SAM + RECGR7.GROUP7;
        SAM[JJ] := 1;
        II := II + 1;
    END;
    CLOSE(FILEGR7);
END;
8:
BEGIN
    ASSIGN(FILEGR8,'G1D08');
    RESET(FILEGR8);
    28:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        REPEAT
            JJ := ROUND(RANDS(IX1) * NOS) ;|
            JJ := RANDOM(NOS+1);
        UNTIL JJ > 0;
        IF SAM[JJ] > 0 THEN
            GOTO 28;
        SEEK(FILEGR8,JJ-1);
        READ(FILEGR8,RECGR8);
        SUM_SAM := SUM_SAM + RECGR8.GROUP8;
        SAM[JJ] := 1;
        II := II + 1;
    END;
    CLOSE(FILEGR8);
END;
9:
BEGIN
    ASSIGN(FILEGR9,'G1D09');
    RESET(FILEGR9);
    29:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }

```

```

BEGIN
  REPEAT
    JJ := ROUND(RANDS(IX1) * NOS) ;|
    JJ := RANDOM(NOS+1);
  UNTIL JJ > 0;
  IF SAM[JJ] > 0 THEN
    GOTO 29;
  SEEK(FILEGR9, JJ-1);
  READ(FILEGR9, RECGR9);
  SUM_SAM := SUM_SAM + RECGR9.GROUP9;
  SAM[JJ] := 1;
  II := II + 1;
END;
CLOSE(FILEGR9);
END;
10:
BEGIN
  ASSIGN(FILEGR10, 'G1D010');
  RESET(FILEGR10);
  30:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    REPEAT
      JJ := ROUND(RANDS(IX1) * NOS) ;|
      JJ := RANDOM(NOS+1);
    UNTIL JJ > 0;
    IF SAM[JJ] > 0 THEN
      GOTO 30;
    SEEK(FILEGR10, JJ-1);
    READ(FILEGR10, RECGR10);
    SUM_SAM := SUM_SAM + RECGR10.GROUP10;
    SAM[JJ] := 1;
    II := II + 1;
  END;
  CLOSE(FILEGR10);
END;
11:
BEGIN
  ASSIGN(FILEGR11, 'G1D011');
  RESET(FILEGR11);
  31:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    REPEAT
      JJ := ROUND(RANDS(IX1) * NOS) ;|
      JJ := RANDOM(NOS+1);
    UNTIL JJ > 0;
    IF SAM[JJ] > 0 THEN
      GOTO 31;
    SEEK(FILEGR11, JJ-1);
    READ(FILEGR11, RECGR11);
    SUM_SAM := SUM_SAM + RECGR11.GROUP11;
    SAM[JJ] := 1;
    II := II + 1;
  END;
END;

```

```

        CLOSE(FILEGR11);
    END;
12:
    BEGIN
        ASSIGN(FILEGR12,'G1D012');
        RESET(FILEGR12);
    32:
        WHILE II <= NO_SAM DO { RANDOM SAMPLE }
        BEGIN
            REPEAT
                JJ := ROUND(RANDS(IX1) * NOS) ;}
                JJ := RANDOM(NOS+1);
            UNTIL JJ > 0;
            IF SAM[JJ] > 0 THEN
                GOTO 32;
            SEEK(FILEGR12, JJ-1);
            READ(FILEGR12, RECGR12);
            SUM_SAM := SUM_SAM + RECGR12.GROUP12;
            SAM[JJ] := 1;
            II := II + 1;
        END;
        CLOSE(FILEGR12);
    END;
END;
        { END CASE }
END;
{*****END RAN_SAM*****}

BEGIN
    SUM_SAM := 0.0;
    FOR N := 1 TO NOG DO
        RAN_SAM(NO_SAM[N], NO[N], N);
    MEAN_STRAT1 := SUM_SAM / I_SAM[I];
    ERROR_STRAT1 := ABS(MEAN_STRAT1 - I_MEAN_NORM) * 100 / I_MEAN_NORM;
    MEAN_STRAT := MEAN_STRAT + MEAN_STRAT1;
    ERROR_STRAT := ERROR_STRAT + ERROR_STRAT1;
    {
        IF (S = 1) OR (S = 2) THEN
            BEGIN
                WRITELN(LST);
                WRITELN(LST, ' ', MEAN_STRAT1:7:4, ' ', ERROR_STRAT1:5:4);
            END;
    }
END;
{*****END RAN_STRAT*****}

PROCEDURE RAN_QUOTA(NOG, S: INTEGER);
VAR
    SS : INTEGER;
    MEAN_QUOTA1, SUM_QSAM : REAL;
    ERROR_QUOTA1 : REAL;
    NN : INTEGER;

    PROCEDURE QUOTA_SAM(NO_SAM, NOS, NN : INTEGER);

    LABEL 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52;
    VAR
        II, MID : INTEGER;

```

```
ADDJ, SUBJ, MAX, MIN, FLAG : INTEGER;
```

```
BEGIN
```

```
IF NO_SAM = 0 THEN
```

```
  EXIT;
```

```
  MIN := 1;
```

```
  MAX := NOS;
```

```
  II := 1;
```

```
  FLAG := 1;
```

```
  CASE NN OF
```

```
    1:
```

```
      BEGIN
```

```
        ASSIGN(FILEGR1, 'G1D01');
```

```
        RESET(FILEGR1);
```

```
        41:
```

```
        WHILE II <= NO_SAM DO { RANDOM SAMPLE }
```

```
        BEGIN
```

```
          CASE FLAG OF
```

```
            1 :
```

```
              BEGIN
```

```
                REPEAT
```

```
                  MID := ROUND(RANDQ(IX2) * MAX) ;}
```

```
                  MID := RANDOM(MAX + 1);
```

```
                UNTIL MID >= MIN;
```

```
                IF MID = MAX THEN
```

```
                  MAX := MAX - 1
```

```
                ELSE
```

```
                IF MID = MIN THEN
```

```
                  MIN := MIN + 1
```

```
                ELSE
```

```
                BEGIN
```

```
                  ADDJ := MID;
```

```
                  SUBJ := MID;
```

```
                  FLAG := 2;
```

```
                END;
```

```
            END;
```

```
            2 :
```

```
              BEGIN
```

```
                ADDJ := ADDJ + 1;
```

```
                IF ADDJ > MAX THEN
```

```
                  BEGIN
```

```
                    MAX := SUBJ-1;
```

```
                    FLAG := 1;
```

```
                    GOTO 41;
```

```
                  END
```

```
                ELSE
```

```
                BEGIN
```

```
                  FLAG := 3;
```

```
                  MID := ADDJ;
```

```
                END;
```

```
            END;
```

```
            3 :
```

```
              BEGIN
```

```
                SUBJ := SUBJ - 1;
```

```
                IF SUBJ = MIN THEN
```

```

        BEGIN
            FLAG := 1;
            MIN := ADDJ+1;
        END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR1,MID-1);
READ(FILEGR1,RECGR1);
SUM_QSAM := SUM_QSAM + RECGR1.GROUP1;
II := II + 1;
END;
CLOSE(FILEGR1);
END;
2:
BEGIN
    ASSIGN(FILEGR2,'G1D02');
    RESET(FILEGR2);
    42:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
                BEGIN
                    REPEAT
                        MID := ROUND(RANDQ(IX2) * MAX) ;|
                        MID := RANDOM(MAX + 1);
                    UNTIL MID >= MIN;
                    IF MID = MAX THEN
                        MAX := MAX - 1
                    ELSE
                        IF MID = MIN THEN
                            MIN := MIN + 1
                        ELSE
                            BEGIN
                                ADDJ := MID;
                                SUBJ := MID;
                                FLAG := 2;
                            END;
                        END;
                    END;
                END;
            2 :
                BEGIN
                    ADDJ := ADDJ + 1;
                    IF ADDJ > MAX THEN
                        BEGIN
                            MAX := SUBJ-1;
                            FLAG := 1;
                            GOTO 42;
                        END
                    ELSE
                        BEGIN
                            FLAG := 3;
                            MID := ADDJ;
                        END
                    END;
                END;
        END;
    END;

```

```

        END;
    END;
3 :
BEGIN
    SUBJ := SUBJ - 1;
    IF SUBJ = MIN THEN
    BEGIN
        FLAG := 1;
        MIN := ADDJ+1;
    END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR2,MID-1);
READ(FILEGR2,RECGR2);
SUM_QSAM := SUM_QSAM + RECGR2.GROUP2;
II := II + 1;
END;
CLOSE(FILEGR2);
END;
3:
BEGIN
    ASSIGN(FILEGR3,'G1D03');
    RESET(FILEGR3);
43:
    WHILE II <= NO_SAM DO [ RANDOM SAMPLE ]
    BEGIN
        CASE FLAG OF
            1 :
                BEGIN
                    REPEAT
                        MID := ROUND(RANDQ(IX2) * MAX) ;|
                        MID := RANDOM(MAX + 1);
                    UNTIL MID >= MIN;
                    IF MID = MAX THEN
                        MAX := MAX - 1
                    ELSE
                        IF MID = MIN THEN
                            MIN := MIN + 1
                        ELSE
                            BEGIN
                                ADDJ := MID;
                                SUBJ := MID;
                                FLAG := 2;
                            END;
                        END;
                    END;
                END;
            2 :
                BEGIN
                    ADDJ := ADDJ + 1;
                    IF ADDJ > MAX THEN
                    BEGIN
                        MAX := SUBJ-1;
                        FLAG := 1;
                    END;
                END;
        END;
    END;

```

```

        GOTO 43;
    END
    ELSE
    BEGIN
        FLAG := 3;
        MID := ADDJ;
    END;
END;
3 :
BEGIN
    SUBJ := SUBJ - 1;
    IF SUBJ = MIN THEN
    BEGIN
        FLAG := 1;
        MIN := ADDJ+1;
    END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR3,MID-1);
READ(FILEGR3,RECGR3);
SUM_QSAM := SUM_QSAM + RECGR3.GROUP3;
II := II + 1;
END;
CLOSE(FILEGR3);
END;
4 :
BEGIN
    ASSIGN(FILEGR4,'G1D04');
    RESET(FILEGR4);
    44:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
            BEGIN
                REPEAT
                    MID := ROUND(RANDQ(IX2) * MAX) ;|
                    MID := RANDOM(MAX + 1);
                UNTIL MID >= MIN;
                IF MID = MAX THEN
                    MAX := MAX - 1
                ELSE
                    IF MID = MIN THEN
                        MIN := MIN + 1
                    ELSE
                        BEGIN
                            ADDJ := MID;
                            SUBJ := MID;
                            FLAG := 2;
                        END;
                    END;
                END;
            END;
        2 :

```



```

BEGIN
  ADDJ := ADDJ + 1;
  IF ADDJ > MAX THEN
    BEGIN
      MAX := SUBJ-1;
      FLAG := 1;
      GOTO 44;
    END
  ELSE
    BEGIN
      FLAG := 3;
      MID := ADDJ;
    END;
  END;
END;
3 :
BEGIN
  SUBJ := SUBJ - 1;
  IF SUBJ = MIN THEN
    BEGIN
      FLAG := 1;
      MIN := ADDJ+1;
    END
  ELSE
    FLAG := 2;
    MID := SUBJ;
  END;
END; { END CASE }
SEEK(FILEGR4,MID-1);
READ(FILEGR4,RECGR4);
SUM_QSAM := SUM_QSAM + RECGR4.GROUP4;
II := II + 1;
END;
CLOSE(FILEGR4);
END;
5:
BEGIN
  ASSIGN(FILEGR5,'G1D05');
  RESET(FILEGR5);
45:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    CASE FLAG OF
      1 :
        BEGIN
          REPEAT
            MID := ROUND(RANDQ(IX2) * MAX) ;
            MID := RANDOM(MAX + 1);
          UNTIL MID >= MIN;
          IF MID = MAX THEN
            MAX := MAX - 1
          ELSE
            IF MID = MIN THEN
              MIN := MIN + 1
            ELSE
              BEGIN

```

```

        ADDJ := MID;
        SUBJ := MID;
        FLAG := 2;
    END;
END;
2 :
BEGIN
    ADDJ := ADDJ + 1;
    IF ADDJ > MAX THEN
    BEGIN
        MAX := SUBJ-1;
        FLAG := 1;
        GOTO 45;
    END
    ELSE
    BEGIN
        FLAG := 3;
        MID := ADDJ;
    END;
END;
3 :
BEGIN
    SUBJ := SUBJ - 1;
    IF SUBJ = MIN THEN
    BEGIN
        FLAG := 1;
        MIN := ADDJ+1;
    END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR5,MID-1);
READ(FILEGR5,RECGR5);
SUM_QSAM := SUM_QSAM + RECGR5.GROUP5;
II := II + 1;
END;
CLOSE(FILEGR5);
END;
6:
BEGIN
    ASSIGN(FILEGR6,'G1D06');
    RESET(FILEGR6);
46:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
                BEGIN
                    REPEAT
                        MID := ROUND(RANDQ(IX2) * MAX) ;}
                        MID := RANDOM(MAX + 1);
                    UNTIL MID >= MIN;
                    IF MID = MAX THEN

```

```

        MAX := MAX - 1
    ELSE
    IF MID = MIN THEN
        MIN := MIN + 1
    ELSE
    BEGIN
        ADDJ := MID;
        SUBJ := MID;
        FLAG := 2;
    END;
END;
2 :
BEGIN
    ADDJ := ADDJ + 1;
    IF ADDJ > MAX THEN
    BEGIN
        MAX := SUBJ-1;
        FLAG := 1;
        GOTO 46;
    END
    ELSE
    BEGIN
        FLAG := 3;
        MID := ADDJ;
    END;
END;
3 :
BEGIN
    SUBJ := SUBJ - 1;
    IF SUBJ = MIN THEN
    BEGIN
        FLAG := 1;
        MIN := ADDJ+1;
    END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR6,MID-1);
READ(FILEGR6,RECGR6);
SUM_QSAM := SUM_QSAM + RECGR6.GROUP6;
II := II + 1;
END;
CLOSE(FILEGR6);
END;
7:
BEGIN
    ASSIGN(FILEGR7,'G1D07');
    RESET(FILEGR7);
47:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :

```

```

BEGIN
  REPEAT
    MID := ROUND(RANDQ(IX2) * MAX) ;|
    MID := RANDOM(MAX + 1);
  UNTIL MID >= MIN;
  IF MID = MAX THEN
    MAX := MAX - 1
  ELSE
    IF MID = MIN THEN
      MIN := MIN + 1
    ELSE
      BEGIN
        ADDJ := MID;
        SUBJ := MID;
        FLAG := 2;
      END;
  END;
2 :
BEGIN
  ADDJ := ADDJ + 1;
  IF ADDJ > MAX THEN
    BEGIN
      MAX := SUBJ-1;
      FLAG := 1;
      GOTO 47;
    END
  ELSE
    BEGIN
      FLAG := 3;
      MID := ADDJ;
    END;
  END;
3 :
BEGIN
  SUBJ := SUBJ - 1;
  IF SUBJ = MIN THEN
    BEGIN
      FLAG := 1;
      MIN := ADDJ+1;
    END
  ELSE
    BEGIN
      FLAG := 2;
      MID := SUBJ;
    END;
  END;
  END; { END CASE }
  SEEK(FILEGR7,MID-1);
  READ(FILEGR7,RECGR7);
  SUM_QSAM := SUM_QSAM + RECGR7.GROUP7;
  II := II + 1;
END;
CLOSE(FILEGR7);
END;
8:
BEGIN
  ASSIGN(FILEGR8,'G1D08');

```

```

RESET(FILEGR8);
48:
WHILE II <= NO_SAM DO { RANDOM SAMPLE }
BEGIN
  CASE FLAG OF
    1 :
      BEGIN
        REPEAT
          MID := ROUND(RANDQ(IX2) * MAX) ;}
          MID := RANDOM(MAX + 1);
        UNTIL MID >= MIN;
        IF MID = MAX THEN
          MAX := MAX - 1
        ELSE
          IF MID = MIN THEN
            MIN := MIN + 1
          ELSE
            BEGIN
              ADDJ := MID;
              SUBJ := MID;
              FLAG := 2;
            END;
          END;
        2 :
          BEGIN
            ADDJ := ADDJ + 1;
            IF ADDJ > MAX THEN
              BEGIN
                MAX := SUBJ-1;
                FLAG := 1;
                GOTO 48;
              END
            ELSE
              BEGIN
                FLAG := 3;
                MID := ADDJ;
              END;
            END;
          3 :
            BEGIN
              SUBJ := SUBJ - 1;
              IF SUBJ = MIN THEN
                BEGIN
                  FLAG := 1;
                  MIN := ADDJ+1;
                END
              ELSE
                BEGIN
                  FLAG := 2;
                  MID := SUBJ;
                END;
              END;
            END;
          END; { END CASE }
        SEEK(FILEGR8,MID-1);
        READ(FILEGR8,RECGR8);
        SUM_QSAM := SUM_QSAM + RECGR8.GROUP8;
        II := II + 1;

```

```

END;
CLOSE(FILEGR8);
END;
9:
BEGIN
  ASSIGN(FILEGR9, 'G1D09');
  RESET(FILEGR9);
  49:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    CASE FLAG OF
      1 :
      BEGIN
        REPEAT
          MID := ROUND(RANDQ(IX2) * MAX) ;}
          MID := RANDOM(MAX + 1);
        UNTIL MID >= MIN;
        IF MID = MAX THEN
          MAX := MAX - 1
        ELSE
          IF MID = MIN THEN
            MIN := MIN + 1
          ELSE
            BEGIN
              ADDJ := MID;
              SUBJ := MID;
              FLAG := 2;
            END;
          END;
        END;
      2 :
      BEGIN
        ADDJ := ADDJ + 1;
        IF ADDJ > MAX THEN
          BEGIN
            MAX := SUBJ-1;
            FLAG := 1;
            GOTO 49;
          END
        ELSE
          BEGIN
            FLAG := 3;
            MID := ADDJ;
          END;
        END;
      3 :
      BEGIN
        SUBJ := SUBJ - 1;
        IF SUBJ = MIN THEN
          BEGIN
            FLAG := 1;
            MIN := ADDJ+1;
          END
        ELSE
          BEGIN
            FLAG := 2;
            MID := SUBJ;
          END;
        END;
      END;
    END;
  END;

```

```

        END;
    END; { END CASE }
    SEEK(FILEGR9,MID-1);
    READ(FILEGR9,RECGR9);
    SUM_QSAM := SUM_QSAM + RECGR9.GROUP9;
    II := II + 1;
END;
CLOSE(FILEGR9);
END;
10:
BEGIN
    ASSIGN(FILEGR10,'G1D010');
    RESET(FILEGR10);
    50:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
            BEGIN
                REPEAT
                    MID := ROUND(RANDQ(IX2) * MAX) ;|
                    MID := RANDOM(MAX + 1);
                UNTIL MID >= MIN;
                IF MID = MAX THEN
                    MAX := MAX - 1
                ELSE
                    IF MID = MIN THEN
                        MIN := MIN + 1
                    ELSE
                        BEGIN
                            ADDJ := MID;
                            SUBJ := MID;
                            FLAG := 2;
                        END;
                    END;
                END;
            2 :
            BEGIN
                ADDJ := ADDJ + 1;
                IF ADDJ > MAX THEN
                    BEGIN
                        MAX := SUBJ-1;
                        FLAG := 1;
                        GOTO 50;
                    END
                ELSE
                    BEGIN
                        FLAG := 3;
                        MID := ADDJ;
                    END;
                END;
            3 :
            BEGIN
                SUBJ := SUBJ - 1;
                IF SUBJ = MIN THEN
                    BEGIN

```

```

        FLAG := 1;
        MIN := ADDJ+1;
    END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR10,MID-1);
READ(FILEGR10,RECGR10);
SUM_QSAM := SUM_QSAM + RECGR10.GROUP10;
II := II + 1;
END;
CLOSE(FILEGR10);
END;
11:
BEGIN
    ASSIGN(FILEGR11,'G1D011');
    RESET(FILEGR11);
    51:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
            BEGIN
                REPEAT
                    MID := ROUND(RANDQ(IX2) * MAX) ;]
                    MID := RANDOM(MAX + 1);
                UNTIL MID >= MIN;
                IF MID = MAX THEN
                    MAX := MAX - 1
                ELSE
                    IF MID = MIN THEN
                        MIN := MIN + 1
                    ELSE
                        BEGIN
                            ADDJ := MID;
                            SUBJ := MID;
                            FLAG := 2;
                        END;
                    END;
                2 :
                BEGIN
                    ADDJ := ADDJ + 1;
                    IF ADDJ > MAX THEN
                        BEGIN
                            MAX := SUBJ-1;
                            FLAG := 1;
                            GOTO 51;
                        END
                    ELSE
                        BEGIN
                            FLAG := 3;
                            MID := ADDJ;
                        END;
                END;
            END;

```



```

END;
3 :
BEGIN
    SUBJ := SUBJ - 1;
    IF SUBJ = MIN THEN
        BEGIN
            FLAG := 1;
            MIN := ADDJ+1;
        END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR11,MID-1);
READ(FILEGR11,RECGR11);
SUM_QSAM := SUM_QSAM + RECGR11.GROUP11;
II := II + 1;
END;
CLOSE(FILEGR11);
END;
12:
BEGIN
    ASSIGN(FILEGR12,'G1D012');
    RESET(FILEGR12);
    52:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
                BEGIN
                    REPEAT
                        MID := ROUND(RANDQ(IX2) * MAX) ;|
                        MID := RANDOM(MAX + 1);
                    UNTIL MID >= MIN;
                    IF MID = MAX THEN
                        MAX := MAX - 1
                    ELSE
                        IF MID = MIN THEN
                            MIN := MIN + 1
                        ELSE
                            BEGIN
                                ADDJ := MID;
                                SUBJ := MID;
                                FLAG := 2;
                            END;
                        END;
                END;
            2 :
                BEGIN
                    ADDJ := ADDJ + 1;
                    IF ADDJ > MAX THEN
                        BEGIN
                            MAX := SUBJ-1;
                            FLAG := 1;
                            GOTO 52;
                        END;
                END;
        END;
    END;

```

```

END
ELSE
BEGIN
    FLAG := 3;
    MID := ADDJ;
END;
END;
3 :
BEGIN
    SUBJ := SUBJ - 1;
    IF SUBJ = MIN THEN
    BEGIN
        FLAG := 1;
        MIN := ADDJ+1;
    END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR12,MID-1);
READ(FILEGR12,RECGR12);
SUM_QSAM := SUM_QSAM + RECGR12.GROUP12;
II := II + 1;
END;
CLOSE(FILEGR12);
END;
END; { END CASE }
{
    CALSTDQT(NO_SAM,SUM_QT,SUM_QT2);
}
{
    WRITELN(LST);
    WRITELN(LST,'SUM    SAM = ',SUM_QSAM:8:2);
}
END;
[*****END RAN_SAM*****]
PROCEDURE QUOTA_MIN(NO_SAM,NOS,NN : INTEGER);
VAR
    II,MID : INTEGER;
    ADDJ,SUBJ,MAX,MIN,FLAG : INTEGER;

BEGIN
    IF NO_SAM = 0 THEN
        EXIT;
    CASE NN OF
        1:
            BEGIN
                ASSIGN(FILEGR1,'G1D01');
                RESET(FILEGR1);
                FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
                BEGIN
                    {
                        SEEK(FILEGR1,II-1);
                        READ(FILEGR1,RECGR1);
                        SUM_QSAM := SUM_QSAM + RECGR1.GROUP1;
                    }
                END;
                CLOSE(FILEGR1);
            END;
        2:

```

```
BEGIN
  ASSIGN(FILEGR2,'G1D02');
  RESET(FILEGR2);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR2,RECGR2);
    SUM_QSAM := SUM_QSAM + RECGR2.GROUP2;
  END;
  CLOSE(FILEGR2);
END;
3:
BEGIN
  ASSIGN(FILEGR3,'G1D03');
  RESET(FILEGR3);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR3,RECGR3);
    SUM_QSAM := SUM_QSAM + RECGR3.GROUP3;
  END;
  CLOSE(FILEGR3);
END;
4:
BEGIN
  ASSIGN(FILEGR4,'G1D04');
  RESET(FILEGR4);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR4,RECGR4);
    SUM_QSAM := SUM_QSAM + RECGR4.GROUP4;
  END;
  CLOSE(FILEGR4);
END;
5:
BEGIN
  ASSIGN(FILEGR5,'G1D05');
  RESET(FILEGR5);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR5,RECGR5);
    SUM_QSAM := SUM_QSAM + RECGR5.GROUP5;
  END;
  CLOSE(FILEGR5);
END;
6:
BEGIN
  ASSIGN(FILEGR6,'G1D06');
  RESET(FILEGR6);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR6,RECGR6);
    SUM_QSAM := SUM_QSAM + RECGR6.GROUP6;
  END;
  CLOSE(FILEGR6);
END;
7:
```

```

BEGIN
  ASSIGN(FILEGR7,'G1D07');
  RESET(FILEGR7);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR7,RECGR7);
    SUM_QSAM := SUM_QSAM + RECGR7.GROUP7;
  END;
  CLOSE(FILEGR7);
END;
8:
BEGIN
  ASSIGN(FILEGR8,'G1D08');
  RESET(FILEGR8);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR8,RECGR8);
    SUM_QSAM := SUM_QSAM + RECGR8.GROUP8;
  END;
  CLOSE(FILEGR8);
END;
9:
BEGIN
  ASSIGN(FILEGR9,'G1D09');
  RESET(FILEGR9);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR9,RECGR9);
    SUM_QSAM := SUM_QSAM + RECGR9.GROUP9;
  END;
  CLOSE(FILEGR9);
END;
10:
BEGIN
  ASSIGN(FILEGR10,'G1D010');
  RESET(FILEGR10);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR10,RECGR10);
    SUM_QSAM := SUM_QSAM + RECGR10.GROUP10;
  END;
  CLOSE(FILEGR10);
END;
11:
BEGIN
  ASSIGN(FILEGR11,'G1D011');
  RESET(FILEGR11);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR11,RECGR11);
    SUM_QSAM := SUM_QSAM + RECGR11.GROUP11;
  END;
  CLOSE(FILEGR11);
END;
12:

```

```

BEGIN
  ASSIGN(FILEGR12,'G1D012');
  RESET(FILEGR12);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    READ(FILEGR12,RECGR12);
    SUM_QSAM := SUM_QSAM + RECGR12.GROUP12;
  END;
  CLOSE(FILEGR12);
END;
END; { END CASE}
{ CALSTDQT(NO_SAM,SUM_QT,SUM_QT2);}
END;
{***** END QUOTA MIN *****}

```

```
PROCEDURE QUOTA_MAX(NO_SAM,NOS,NN : INTEGER);
```

```
VAR
```

```
  III,II,MID : INTEGER;
```

```
  ADDJ,SUBJ,MAX,MIN,FLAG : INTEGER;
```

```
BEGIN
```

```
  IF NO_SAM = 0 THEN
```

```
    EXIT;
```

```
  SUM_QT := 0.0;
```

```
  SUM_QT2 := 0.0;}
```

```
  III := NOS;
```

```
  CASE NN OF
```

```
    1:
```

```
    BEGIN
```

```
      ASSIGN(FILEGR1,'G1D01');
```

```
      RESET(FILEGR1);
```

```
      FOR II := 1 TO NO_SAM DO
```

```
      BEGIN
```

```
        III := III - 1;
```

```
        SEEK(FILEGR1,III);
```

```
        READ(FILEGR1,RECGR1);
```

```
        SUM_QSAM := SUM_QSAM + RECGR1.GROUP1;
```

```
      END;
```

```
      CLOSE(FILEGR1);
```

```
    END;
```

```
    2:
```

```
    BEGIN
```

```
      ASSIGN(FILEGR2,'G1D02');
```

```
      RESET(FILEGR2);
```

```
      FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
```

```
      BEGIN
```

```
        III := III - 1;
```

```
        SEEK(FILEGR2,III);
```

```
        READ(FILEGR2,RECGR2);
```

```
        SUM_QSAM := SUM_QSAM + RECGR2.GROUP2;
```

```
      END;
```

```
      CLOSE(FILEGR2);
```

```
    END;
```

```
    3:
```

```

BEGIN
  ASSIGN(FILEGR3,'G1D03');
  RESET(FILEGR3);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    III := III - 1;
    SEEK(FILEGR3,III);
    READ(FILEGR3,RECGR3);
    SUM_QSAM := SUM_QSAM + RECGR3.GROUP3;
  END;
  CLOSE(FILEGR3);
END;
4:
BEGIN
  ASSIGN(FILEGR4,'G1D04');
  RESET(FILEGR4);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    III := III - 1;
    SEEK(FILEGR4,III);
    READ(FILEGR4,RECGR4);
    SUM_QSAM := SUM_QSAM + RECGR4.GROUP4;
  END;
  CLOSE(FILEGR4);
END;
5:
BEGIN
  ASSIGN(FILEGR5,'G1D05');
  RESET(FILEGR5);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    III := III - 1;
    SEEK(FILEGR5,III);
    READ(FILEGR5,RECGR5);
    SUM_QSAM := SUM_QSAM + RECGR5.GROUP5;
  END;
  CLOSE(FILEGR5);
END;
6:
BEGIN
  ASSIGN(FILEGR6,'G1D06');
  RESET(FILEGR6);
  FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    III := III - 1;
    SEEK(FILEGR6,III);
    READ(FILEGR6,RECGR6);
    SUM_QSAM := SUM_QSAM + RECGR6.GROUP6;
  END;
  CLOSE(FILEGR6);
END;
7:
BEGIN
  ASSIGN(FILEGR7,'G1D07');
  RESET(FILEGR7);

```

```

FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
BEGIN
    III := III - 1;
    SEEK(FILEGR7,III);
    READ(FILEGR7,RECGR7);
    SUM_QSAM := SUM_QSAM + RECGR7.GROUP7;
END;
CLOSE(FILEGR7);
END;
8:
BEGIN
    ASSIGN(FILEGR8,'G1D08');
    RESET(FILEGR8);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        III := III - 1;
        SEEK(FILEGR8,III);
        READ(FILEGR8,RECGR8);
        SUM_QSAM := SUM_QSAM + RECGR8.GROUP8;
    END;
    CLOSE(FILEGR8);
END;
9:
BEGIN
    ASSIGN(FILEGR9,'G1D09');
    RESET(FILEGR9);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        III := III - 1;
        SEEK(FILEGR9,III);
        READ(FILEGR9,RECGR9);
        SUM_QSAM := SUM_QSAM + RECGR9.GROUP9;
    END;
    CLOSE(FILEGR9);
END;
10:
BEGIN
    ASSIGN(FILEGR10,'G1D010');
    RESET(FILEGR10);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        III := III - 1;
        SEEK(FILEGR10,III);
        READ(FILEGR10,RECGR10);
        SUM_QSAM := SUM_QSAM + RECGR10.GROUP10;
    END;
    CLOSE(FILEGR10);
END;
11:
BEGIN
    ASSIGN(FILEGR11,'G1D011');
    RESET(FILEGR11);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        III := III - 1;

```

```

        SEEK(FILEGR11,III);
        READ(FILEGR11,RECGR11);
        SUM_QSAM := SUM_QSAM + RECGR11.GROUP11;
    END;
    CLOSE(FILEGR11);
END;
12:
BEGIN
    ASSIGN(FILEGR12,'G1D012');
    RESET(FILEGR12);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        III := III - 1;
        SEEK(FILEGR12,III);
        READ(FILEGR12,RECGR12);
        SUM_QSAM := SUM_QSAM + RECGR12.GROUP12;
    END;
    CLOSE(FILEGR12);
END;
END; { END CASE}
END;

BEGIN
    SUM_QSAM := 0.0;
    IF S = 1 THEN
        FOR NN := 1 TO NOG DO
            QUOTA_MIN(NO_SAM[NN],NO[NN],NN)
        ELSE
            IF S = 2 THEN
                FOR NN := 1 TO NOG DO
                    QUOTA_MAX(NO_SAM[NN],NO[NN],NN)
                ELSE
                    FOR NN := 1 TO NOG DO
                        QUOTA_SAM(NO_SAM[NN],NO[NN],NN);
                    MEAN_QUOTA1 := SUM_QSAM / I_SAM[I];
                    ERROR_QUOTA1 := ABS(MEAN_QUOTA1 - I_MEAN_NORM) * 100 / I_MEAN_NORM;
                    WRITELN;
                    WRITE(' ',MEAN_QUOTA1:7:2,' ',ERROR_QUOTA1:5:2);}
                    IF (S = 1) OR (S = 2) THEN
                        BEGIN
                            WRITELN(LST);
                            WRITELN(LST,' ',MEAN_QUOTA1:7:4,' ',ERROR_QUOTA1:5:4);
                        END
                    ELSE
                        BEGIN
                            MEAN_QUOTA := MEAN_QUOTA + MEAN_QUOTA1;
                            ERROR_QUOTA := ERROR_QUOTA + ERROR_QUOTA1;
                        END;
                    END;
                END;
            [*****END QUOTA_SAM*****]

PROCEDURE SAM_LEVEL(G:INTEGER);
VAR

```



```

S,NO_TMP, JJ, J : INTEGER;
SUM_GROUP, SUM_GROUP2, MEAN_GROUP, STD_GROUP, STD_GROUP2 : REAL;

PROCEDURE CALCV(NN: INTEGER);
VAR CVG : REAL;
BEGIN
  IF I = 1 THEN
    BEGIN
      MEAN_GROUP := SUM_GROUP / NO[NN];
      STD_GROUP := (SUM_GROUP2 - (NO[NN] * SQR(MEAN_GROUP))) / (NO[NN] - 1);
      STD_GROUP2 := SQR(STD_GROUP);
      CVG := (STD_GROUP2 / MEAN_GROUP) * 100;
      WRITELN(LST, 'POP STRATIFIED GROUP ', NN, ' MEAN = ', MEAN_GROUP:8:2,
        ', STD = ', STD_GROUP2:5:2, ' C.V. = ', CVG:3:2);
    END;
  END;

PROCEDURE CALSUM;
BEGIN
  IF I = 1 THEN
    BEGIN
      SUM_GROUP := SUM_GROUP + RECX.Y;
      SUM_GROUP2 := SUM_GROUP2 + SQR(RECX.Y);
    END;
  END;

BEGIN
  MEAN_STRAT := 0.0;
  MEAN_QUOTA := 0.0;
  ERROR_STRAT := 0.0;
  ERROR_QUOTA := 0.0;
  FOR S := 1 TO 12 DO
    BEGIN
      NO[S] := 0;
    END;
  ASSIGN(FILEXY, 'XY1D01');
  RESET(FILEXY);
  CASE G OF
    1:
      BEGIN
        J := 1;
        SUM_GROUP := 0.0;
        SUM_GROUP2 := 0.0;
        WHILE NOT EOF(FILEXY) DO
          BEGIN
            READ (FILEXY, RECX);
            S := QTRAN12;
            IF S = 1 THEN
              BEGIN
                NO[S] := NO[S] + 1;
                YG[J] := RECX.Y;
                J := J + 1;
                CALSUM;
              END;
            END;
          END;
        END;
      END;
    END;
  END;

```

```

quicksort(YG,1,NO[1]);
ASSIGN(FILEGR1,'G1D01');
REWRITE(FILEGR1);
FOR J := 1 TO NO[1] DO
BEGIN
    RECGR1.GROUP1 := YG[J];
    WRITE(FILEGR1,RECGR1);
END;
CLOSE(FILEGR1);
CALCV(1);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN12;
    IF S = 2 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[2]);
ASSIGN(FILEGR2,'G1D02');
REWRITE(FILEGR2);
FOR J := 1 TO NO[2] DO
BEGIN
    RECGR2.GROUP2 := YG[J];
    WRITE(FILEGR2,RECGR2);
END;
CLOSE(FILEGR2);
CALCV(2);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN12;
    IF S = 3 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[3]);
ASSIGN(FILEGR3,'G1D03');
REWRITE(FILEGR3);

```

```

FOR J := 1 TO NO[3] DO
BEGIN
    RECGR3.GROUP3 := YG[J];
    WRITE(FILEGR3,RECGR3);
END;
CLOSE(FILEGR3);
CALCV(3);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN12;
    IF S = 4 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[4]);
ASSIGN(FILEGR4,'G1D04');
REWRITE(FILEGR4);
FOR J := 1 TO NO[4] DO
BEGIN
    RECGR4.GROUP4 := YG[J];
    WRITE(FILEGR4,RECGR4);
END;
CLOSE(FILEGR4);
CALCV(4);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN12;
    IF S = 5 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[5]);
ASSIGN(FILEGR5,'G1D05');
REWRITE(FILEGR5);
FOR J := 1 TO NO[5] DO
BEGIN
    RECGR5.GROUP5 := YG[J];

```

```

WRITE(FILEGR5,RECGR5);
END;
CLOSE(FILEGR5);
CALCV(5);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
  READ (FILEXY,RECY);
  S := QTRAN12;
  IF S = 6 THEN
  BEGIN
    NO[S] := NO[S] + 1;
    YG[J] := RECY.Y;
    J := J + 1;
    CALSUM;
  END;
END;
quicksort(YG,1,NO[6]);
ASSIGN(FILEGR6,'G1D06');
REWRITE(FILEGR6);
FOR J := 1 TO NO[6] DO
BEGIN
  RECGR6.GROUP6 := YG[J];
  WRITE(FILEGR6,RECGR6);
END;
CLOSE(FILEGR6);
CALCV(6);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
  READ (FILEXY,RECY);
  S := QTRAN12;
  IF S = 7 THEN
  BEGIN
    NO[S] := NO[S] + 1;
    YG[J] := RECY.Y;
    J := J + 1;
    CALSUM;
  END;
END;
quicksort(YG,1,NO[7]);
ASSIGN(FILEGR7,'G1D07');
REWRITE(FILEGR7);
FOR J := 1 TO NO[7] DO
BEGIN
  RECGR7.GROUP7 := YG[J];
  WRITE(FILEGR7,RECGR7);
END;
CLOSE(FILEGR7);

```

```

CALCV(7);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
  READ (FILEXY,RECY);
  S := QTRAN12;
  IF S = 8 THEN
  BEGIN
    NO[S] := NO[S] + 1;
    YG[J] := RECY.Y;
    J := J + 1;
    CALSUM;
  END;
END;
quicksort(YG,1,NO[8]);
ASSIGN(FILEGR8,'G1D08');
REWRITE(FILEGR8);
FOR J := 1 TO NO[8] DO
BEGIN
  RECGR8.GROUP8 := YG[J];
  WRITE(FILEGR8,RECGR8);
END;
CLOSE(FILEGR8);
CALCV(8);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
  READ (FILEXY,RECY);
  S := QTRAN12;
  IF S = 9 THEN
  BEGIN
    NO[S] := NO[S] + 1;
    YG[J] := RECY.Y;
    J := J + 1;
    CALSUM;
  END;
END;
quicksort(YG,1,NO[9]);
ASSIGN(FILEGR9,'G1D09');
REWRITE(FILEGR9);
FOR J := 1 TO NO[9] DO
BEGIN
  RECGR9.GROUP9 := YG[J];
  WRITE(FILEGR9,RECGR9);
END;
CLOSE(FILEGR9);
CALCV(9);
J := 1;
SUM_GROUP := 0.0;

```

```

SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
  READ (FILEXY,RECY);
  S := QTRAN12;
  IF S = 10 THEN
  BEGIN
    NO[S] := NO[S] + 1;
    YG[J] := RECY.Y;
    J := J + 1;
    CALSUM;
  END;
END;
quicksort(YG,1,NO[10]);
ASSIGN(FILEGR10,'G1D010');
REWRITE(FILEGR10);
FOR J := 1 TO NO[10] DO
BEGIN
  RECGR10.GROUP10 := YG[J];
  WRITE(FILEGR10,RECGR10);
END;
CLOSE(FILEGR10);
CALCV(10);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
  READ (FILEXY,RECY);
  S := QTRAN12;
  IF S = 11 THEN
  BEGIN
    NO[S] := NO[S] + 1;
    YG[J] := RECY.Y;
    J := J + 1;
    CALSUM;
  END;
END;
quicksort(YG,1,NO[11]);
ASSIGN(FILEGR11,'G1D011');
REWRITE(FILEGR11);
FOR J := 1 TO NO[11] DO
BEGIN
  RECGR11.GROUP11 := YG[J];
  WRITE(FILEGR11,RECGR11);
END;
CLOSE(FILEGR11);
CALCV(11);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO

```

```

BEGIN
  READ (FILEXY,RECXY);
  S := QTRAN12;
  IF S = 12 THEN
    BEGIN
      NO[S] := NO[S] + 1;
      YG[J] := RECXY.Y;
      J := J + 1;
      CALSUM;
    END;
  END;
  quicksort(YG,1,NO[12]);
  ASSIGN(FILEGR12,'G1D012');
  REWRITE(FILEGR12);
  FOR J := 1 TO NO[12] DO
    BEGIN
      RECGR12.GROUP12 := YG[J];
      WRITE(FILEGR12,RECGR12);
    END;
  CLOSE(FILEGR12);
  CALCV(12);
  PRTG(12);
  NO_TMP := 0;
  FOR J := 1 TO 11 DO
    BEGIN
      NO_SAM[J] := ROUND((NO[J] / I_COUNT)*I_SAM[I] );
      NO_TMP := NO_TMP + NO_SAM[J];
    END;
  NO_SAM[12] := I_SAM[I] - NO_TMP;
  writeln(lst);
  WRITELN(LST,'MEAN STRAT 12 GROUP(X1 =4,X2 = 3) ':40);
  FOR S := 1 TO S_COUNT DO
    RAN_STRAT(12,S);
  WRITE(LST,'MEAN QUOTA 12 GROUP(X1 =4,X2 = 3) ':40);
  FOR S := 1 TO S_COUNT DO
    RAN_QUOTA(12,S);
  END;
2:
BEGIN
  J := 1;
  SUM_GROUP := 0.0;
  SUM_GROUP2 := 0.0;
  WHILE NOT EOF(FILEXY) DO
    BEGIN
      READ (FILEXY,RECXY);
      S := QTRAN7;
      IF S = 1 THEN
        BEGIN
          NO[S] := NO[S] + 1;
          YG[J] := RECXY.Y;
          J := J + 1;
          CALSUM;
        END;
      END;
    quicksort(YG,1,NO[1]);

```

```

ASSIGN(FILEGR1,'G1D01');
REWRITE(FILEGR1);
FOR J := 1 TO NO[1] DO
BEGIN
    RECGR1.GROUP1 := YG[J];
    WRITE(FILEGR1,RECGR1);
END;
CLOSE(FILEGR1);
CALCV(1);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN7;
    IF S = 2 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[2]);
ASSIGN(FILEGR2,'G1D02');
REWRITE(FILEGR2);
FOR J := 1 TO NO[2] DO
BEGIN
    RECGR2.GROUP2 := YG[J];
    WRITE(FILEGR2,RECGR2);
END;
CLOSE(FILEGR2);
CALCV(2);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN7;
    IF S = 3 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[3]);
ASSIGN(FILEGR3,'G1D03');
REWRITE(FILEGR3);
FOR J := 1 TO NO[3] DO

```



```

BEGIN
    RECGR3.GROUP3 := YG[J];
    WRITE(FILEGR3,RECGR3);
END;
CLOSE(FILEGR3);
CALCV(3);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN7;
    IF S = 4 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[4]);
ASSIGN(FILEGR4,'G1D04');
REWRITE(FILEGR4);
FOR J := 1 TO NO[4] DO
BEGIN
    RECGR4.GROUP4 := YG[J];
    WRITE(FILEGR4,RECGR4);
END;
CLOSE(FILEGR4);
CALCV(4);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN7;
    IF S = 5 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[5]);
ASSIGN(FILEGR5,'G1D05');
REWRITE(FILEGR5);
FOR J := 1 TO NO[5] DO
BEGIN
    RECGR5.GROUP5 := YG[J];
    WRITE(FILEGR5,RECGR5);

```

```

END;
CLOSE(FILEGR5);
CALCV(5);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN7;
    IF S = 6 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[6]);
ASSIGN(FILEGR6,'G1D06');
REWRITE(FILEGR6);
FOR J := 1 TO NO[6] DO
BEGIN
    RECGR6.GROUP6 := YG[J];
    WRITE(FILEGR6,RECGR6);
END;
CLOSE(FILEGR6);
CALCV(6);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN7;
    IF S = 7 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[7]);
ASSIGN(FILEGR7,'G1D07');
REWRITE(FILEGR7);
FOR J := 1 TO NO[7] DO
BEGIN
    RECGR7.GROUP7 := YG[J];
    WRITE(FILEGR7,RECGR7);
END;
CLOSE(FILEGR7);
CALCV(7);

```

```

NO_TMP := 0;
FOR J := 1 TO 6 DO
BEGIN
    NO_SAM[J] := ROUND((NO[J] / I_COUNT)*I_SAM[I] );
    NO_TMP := NO_TMP + NO_SAM[J];
END;
NO_SAM[7] := I_SAM[I] - NO_TMP;
WRITELN(LST);
WRITELN(LST,'MEAN STRAT 7 GROUP ':40);
FOR S := 1 TO S_COUNT DO
    RAN_STRAT(7,S);
WRITE(LST,'MEAN QUOTA 7 GROUP ':40);
FOR S := 1 TO S_COUNT DO
    RAN_QUOTA(7,S);
END;
3:
BEGIN .
    J := 1;
    SUM_GROUP := 0.0;
    SUM_GROUP2 := 0.0;
    WHILE NOT EOF(FILEXY) DO
    BEGIN
        READ (FILEXY,RECY);
        S := QTRAN3;
        IF S = 1 THEN
        BEGIN
            NO[S] := NO[S] + 1;
            YG[J] := RECY.Y;
            J := J + 1;
            CALSUM;
        END;
    END;
    quicksort(YG,1,NO[1]);
    ASSIGN(FILEGR1,'G1D01');
    REWRITE(FILEGR1);
    FOR J := 1 TO NO[1] DO
    BEGIN
        RECGR1.GROUP1 := YG[J];
        WRITE(FILEGR1,RECGR1);
    END;
    CLOSE(FILEGR1);
    CALCV(1);
    J := 1;
    SUM_GROUP := 0.0;
    SUM_GROUP2 := 0.0;
    RESET (FILEXY);
    WHILE NOT EOF(FILEXY) DO
    BEGIN
        READ (FILEXY,RECY);
        S := QTRAN3;
        IF S = 2 THEN
        BEGIN
            NO[S] := NO[S] + 1;
            YG[J] := RECY.Y;
            J := J + 1;
        END;
    END;

```

```

        CALSUM;
    END;
END;
quicksort(YG,1,NO[2]);
ASSIGN(FILEGR2,'G1D02');
REWRITE(FILEGR2);
FOR J := 1 TO NO[2] DO
BEGIN
    RECGR2.GROUP2 := YG[J];
    WRITE(FILEGR2,RECGR2);
END;
CLOSE(FILEGR2);
CALCV(2);
J := 1;
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
RESET (FILEXY);
WHILE NOT EOF(FILEXY) DO
BEGIN
    READ (FILEXY,RECY);
    S := QTRAN3;
    IF S = 3 THEN
    BEGIN
        NO[S] := NO[S] + 1;
        YG[J] := RECY.Y;
        J := J + 1;
        CALSUM;
    END;
END;
quicksort(YG,1,NO[3]);
ASSIGN(FILEGR3,'G1D03');
REWRITE(FILEGR3);
FOR J := 1 TO NO[3] DO
BEGIN
    RECGR3.GROUP3 := YG[J];
    WRITE(FILEGR3,RECGR3);
END;
CLOSE(FILEGR3);
CALCV(3);
NO_TMP := 0;
FOR J := 1 TO 2 DO
BEGIN
    NO_SAM[J] := ROUND((NO[J] / I_COUNT)*I_SAM[I]);
    NO_TMP := NO_TMP + NO_SAM[J];
END;
NO_SAM[3] := I_SAM[I] - NO_TMP;
Writeln(LST);
Writeln(LST,'MEAN STRAT 3 GROUP ':40);
FOR S := 1 TO S_COUNT DO
    RAN_STRAT(3,S);
WRITE(LST,'MEAN QUOTA 3 GROUP ':40);
FOR S := 1 TO S_COUNT DO
    RAN_QUOTA(3,S);
END;
END;
END;  {** END CASE **}

```

```

CLOSE(FILEXY);
MEAN_STRAT := MEAN_STRAT / S_COUNT;
[
  Writeln;
  Writeln('MEAN_STRAT ',G,' = ',MEAN_STRAT:7:2);}
ERROR_STRAT := ERROR_STRAT / S_COUNT;
[
  Writeln('ERROR_STRAT ',G,' = ',ERROR_STRAT:5:2);}
MEAN_QUOTA := MEAN_QUOTA / (S_COUNT - 2);
ERROR_QUOTA := ERROR_QUOTA / (S_COUNT - 2);
[
  Writeln('MEAN_QUOTA ',G,' = ',MEAN_QUOTA:7:2);
  Writeln('ERROR_QUOTA ',G,' = ',ERROR_QUOTA:5:2);}
Writeln(LST);
WRITE(LST,'MEAN_STRAT ',G,' = ',MEAN_STRAT:9:4);
WRITE(LST,' ERROR_STRAT ',G,' = ',ERROR_STRAT:5:4);
Writeln(LST);
WRITE(LST,'MEAN_QUOTA ',G,' = ',MEAN_QUOTA:9:4);
Writeln(LST,' ERROR_QUOTA ',G,' = ',ERROR_QUOTA:5:4);
END;

BEGIN  {MAIN}
IX := 65479 {65539} {7257371} {9753451} {START VALUE PROGRAM};
IX1 := 65479;
IX2 := 65479;
KK := 0;
I_STD_NORM[1] := 200;
CORR_X1_X2[1] := 0.1;
CORR_X1_Y[1] := 0.1;
CORR_X2_Y[1] := 0.1;
I_STD_NORM[2] := 200;
CORR_X1_X2[2] := 0.625;
CORR_X1_Y[2] := 0.5;
CORR_X2_Y[2] := 0.5;
I_STD_NORM[3] := 200;
CORR_X1_X2[3] := 0.1;
CORR_X1_Y[3] := 0.9;
CORR_X2_Y[3] := 0.5;
I_STD_NORM[4] := 500;
CORR_X1_X2[4] := 0.1;
CORR_X1_Y[4] := 0.1;
CORR_X2_Y[4] := 0.1;
I_STD_NORM[5] := 500;
CORR_X1_X2[5] := 0.625;
CORR_X1_Y[5] := 0.5;
CORR_X2_Y[5] := 0.5;
I_STD_NORM[6] := 500;
CORR_X1_X2[6] := 0.1;
CORR_X1_Y[6] := 0.9;
CORR_X2_Y[6] := 0.5;
I_STD_NORM[7] := 1000;
CORR_X1_X2[7] := 0.1;
CORR_X1_Y[7] := 0.1;
CORR_X2_Y[7] := 0.1;
I_STD_NORM[8] := 1000;
CORR_X1_X2[8] := 0.625;
CORR_X1_Y[8] := 0.5;
CORR_X2_Y[8] := 0.5;

```

```

I_STD_NORM[9] := 1000;
CORR_X1_X2[9] := 0.1;
CORR_X1_Y[9] := 0.9;
CORR_X2_Y[9] := 0.5;
I_STD_NORM[10] := 2000;
CORR_X1_X2[10] := 0.1;
CORR_X1_Y[10] := 0.1;
CORR_X2_Y[10] := 0.1;
I_STD_NORM[11] := 2000;
CORR_X1_X2[11] := 0.625;
CORR_X1_Y[11] := 0.5;
CORR_X2_Y[11] := 0.5;
I_STD_NORM[12] := 2000;
CORR_X1_X2[12] := 0.1;
CORR_X1_Y[12] := 0.9;
CORR_X2_Y[12] := 0.5;
I_SAM[1] := 250;
I_SAM[2] := 500;
I_SAM[3] := 750;
CLRSCR;
FOR K:= 1 TO K_COUNT DO                                {** REPEAT SIMULATION **}
BEGIN
  CVP := (I_STD_NORM[K] / I_MEAN_NORM ) * 100;
  WRITELN(LST);
  WRITELN(LST);
  WRITELN(LST, '***** POPULATION ', K, ' *****');
  WRITE(LST, 'I_COUNT = ', I_COUNT:4);
  WRITE(LST, ' ', 'TOTAL REPEAT = ', K_COUNT:4);
  WRITE(LST, ' ', 'TOTAL RANDOM = ', S_COUNT:4);
  WRITELN(LST);
  WRITE(LST, 'MEAN = ', I_MEAN_NORM:5:2);
  WRITE(LST, ' ', 'STD = ', I_STD_NORM[K]:5:2);
  WRITE(LST, ' ', 'C.V. = ', CVP:3:2);
  WRITELN(LST);
  WRITE(LST, 'CORR_X1_X2 = ', CORR_X1_X2[K]:4:2);
  WRITE(LST, ' ', 'CORR_X1_Y = ', CORR_X1_Y[K]:4:2);
  WRITE(LST, ' ', 'CORR_X2_Y = ', CORR_X2_Y[K]:4:2);
  WRITELN(LST);
  SIM_NORM;
  PRCDATA;
  TRANX;                                               {** TRANSFER X1=1,2,3,4 : X2=1,2,3 **}
  FOR I := 1 TO 3 DO
  BEGIN
    WRITELN(LST);
    WRITELN(LST, '**** I_SAM = ', I_SAM[I]:4, ' ****');
    WRITELN(LST);
    FOR G := 1 TO 3 DO
    BEGIN
      SAM_LEVEL(G);                                     {** SAMPLING FOR STRATIFIED & QUOTA **}
    END;
    PRTG(G);
  END;
  IX := IX + K;
END; { END LOOP K := 1 TO K_COUNT }
END.

```

```

REPEAT
  MID := RANDOM(MAX + 1);
UNTIL MID >= MIN;
IF MID = MAX THEN
  MAX := MAX - 1
ELSE
  IF MID = MIN THEN
    MIN := MIN + 1
  ELSE
    BEGIN
      ADDJ := MID;
      SUBJ := MID;
      FLAG := 2;
    END;
  END;
2 :
BEGIN
  ADDJ := ADDJ + 1;
  IF ADDJ > MAX THEN
    BEGIN
      MAX := SUBJ-1;
      FLAG := 1;
      GOTO 47;
    END
  ELSE
    BEGIN
      FLAG := 3;
      MID := ADDJ;
    END;
  END;
3 :
BEGIN
  SUBJ := SUBJ - 1;
  IF SUBJ = MIN THEN
    BEGIN
      FLAG := 1;
      MIN := ADDJ+1;
    END
  ELSE
    BEGIN
      FLAG := 2;
      MID := SUBJ;
    END;
  END;
END; { END CASE }
SEEK(FILEGR7,MID-1);
READ(FILEGR7,RECGR7);
SUM_QSAM := SUM_QSAM + RECGR7.GROUP7;
II := II + 1;
END;
CLOSE(FILEGR7);
END;
8:
BEGIN
  ASSIGN(FILEGR8,'G1D08');
  RESET(FILEGR8);
48:

```

```

WHILE II <= NO_SAM DO { RANDOM SAMPLE }
BEGIN
  CASE FLAG OF
    1 :
      BEGIN
        REPEAT
          MID := RANDOM(MAX + 1);
        UNTIL MID >= MIN;
        IF MID = MAX THEN
          MAX := MAX - 1
        ELSE
          IF MID = MIN THEN
            MIN := MIN + 1
          ELSE
            BEGIN
              ADDJ := MID;
              SUBJ := MID;
              FLAG := 2;
            END;
          END;
        2 :
          BEGIN
            ADDJ := ADDJ + 1;
            IF ADDJ > MAX THEN
              BEGIN
                MAX := SUBJ-1;
                FLAG := 1;
                GOTO 48;
              END
            ELSE
              BEGIN
                FLAG := 3;
                MID := ADDJ;
              END;
            END;
          3 :
            BEGIN
              SUBJ := SUBJ - 1;
              IF SUBJ = MIN THEN
                BEGIN
                  FLAG := 1;
                  MIN := ADDJ+1;
                END
              ELSE
                BEGIN
                  FLAG := 2;
                  MID := SUBJ;
                END;
              END;
            END; { END CASE }
            SEEK(FILEGR8,MID-1);
            READ(FILEGR8,RECGR8);
            SUM_QSAM := SUM_QSAM + RECGR8.GROUP8;
            II := II + 1;
          END;
        CLOSE(FILEGR8);
      END;

```



```

9:
BEGIN
  ASSIGN(FILEGR9, 'G1D09');
  RESET(FILEGR9);
  49:
  WHILE II <= NO_SAM DO { RANDOM SAMPLE }
  BEGIN
    CASE FLAG OF
      1 :
      BEGIN
        REPEAT
          MID := RANDOM(MAX + 1);
        UNTIL MID >= MIN;
        IF MID = MAX THEN
          MAX := MAX - 1
        ELSE
          IF MID = MIN THEN
            MIN := MIN + 1
          ELSE
            BEGIN
              ADDJ := MID;
              SUBJ := MID;
              FLAG := 2;
            END;
          END;
        END;
      2 :
      BEGIN
        ADDJ := ADDJ + 1;
        IF ADDJ > MAX THEN
          BEGIN
            MAX := SUBJ-1;
            FLAG := 1;
            GOTO 49;
          END
        ELSE
          BEGIN
            FLAG := 3;
            MID := ADDJ;
          END;
        END;
      3 :
      BEGIN
        SUBJ := SUBJ - 1;
        IF SUBJ = MIN THEN
          BEGIN
            FLAG := 1;
            MIN := ADDJ+1;
          END
        ELSE
          BEGIN
            FLAG := 2;
            MID := SUBJ;
          END;
        END;
      END; { END CASE }
    SEEK(FILEGR9, MID-1);
    READ(FILEGR9, RECGR9);
  END;

```

```

SUM_QSAM := SUM_QSAM + RECGR9.GROUP9;
II := II + 1;
END;
CLOSE(FILEGR9);
END;
10:
BEGIN
ASSIGN(FILEGR10,'G1D010');
RESET(FILEGR10);
50:
WHILE II <= NO_SAM DO { RANDOM SAMPLE }
BEGIN
CASE FLAG OF
1 :
BEGIN
REPEAT
MID := RANDOM(MAX + 1);
UNTIL MID >= MIN;
IF MID = MAX THEN
MAX := MAX - 1
ELSE
IF MID = MIN THEN
MIN := MIN + 1
ELSE
BEGIN
ADDJ := MID;
SUBJ := MID;
FLAG := 2;
END;
END;
2 :
BEGIN
ADDJ := ADDJ + 1;
IF ADDJ > MAX THEN
BEGIN
MAX := SUBJ-1;
FLAG := 1;
GOTO 50;
END
ELSE
BEGIN
FLAG := 3;
MID := ADDJ;
END;
END;
3 :
BEGIN
SUBJ := SUBJ - 1;
IF SUBJ = MIN THEN
BEGIN
FLAG := 1;
MIN := ADDJ+1;
END
ELSE
FLAG := 2;

```

```

                MID := SUBJ;
            END;
        END; { END CASE }
        SEEK(FILEGR10,MID-1);
        READ(FILEGR10,RECGR10);
        SUM_QSAM := SUM_QSAM + RECGR10.GROUP10;
        II := II + 1;
    END;
    CLOSE(FILEGR10);
END;
11:
BEGIN
    ASSIGN(FILEGR11,'G1D011');
    RESET(FILEGR11);
    51:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
            BEGIN
                REPEAT
                    MID := RANDOM(MAX + 1);
                UNTIL MID >= MIN;
                IF MID = MAX THEN
                    MAX := MAX - 1
                ELSE
                    IF MID = MIN THEN
                        MIN := MIN + 1
                    ELSE
                        BEGIN
                            ADDJ := MID;
                            SUBJ := MID;
                            FLAG := 2;
                        END;
                END;
            END;
            2 :
            BEGIN
                ADDJ := ADDJ + 1;
                IF ADDJ > MAX THEN
                    BEGIN
                        MAX := SUBJ-1;
                        FLAG := 1;
                        GOTO 51;
                    END
                ELSE
                    BEGIN
                        FLAG := 3;
                        MID := ADDJ;
                    END;
            END;
            3 :
            BEGIN
                SUBJ := SUBJ - 1;
                IF SUBJ = MIN THEN
                    BEGIN

```

```

        FLAG := 1;
        MIN := ADDJ+1;
    END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR11,MID-1);
READ(FILEGR11,RECGR11);
SUM_QSAM := SUM_QSAM + RECGR11.GROUP11;
II := II + 1;
END;
CLOSE(FILEGR11);
END;
12:
BEGIN
    ASSIGN(FILEGR12,'G1D012');
    RESET(FILEGR12);
52:
    WHILE II <= NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        CASE FLAG OF
            1 :
                BEGIN
                    REPEAT
                        MID := RANDOM(MAX + 1);
                    UNTIL MID >= MIN;
                    IF MID = MAX THEN
                        MAX := MAX - 1
                    ELSE
                        IF MID = MIN THEN
                            MIN := MIN + 1
                        ELSE
                            BEGIN
                                ADDJ := MID;
                                SUBJ := MID;
                                FLAG := 2;
                            END;
                        END;
                    2 :
                        BEGIN
                            ADDJ := ADDJ + 1;
                            IF ADDJ > MAX THEN
                                BEGIN
                                    MAX := SUBJ-1;
                                    FLAG := 1;
                                    GOTO 52;
                                END
                            END
                        ELSE
                            BEGIN
                                FLAG := 3;
                                MID := ADDJ;
                            END;
                        END;
                    END;
                END;
            END;
END;

```

```

3 :
BEGIN
    SUBJ := SUBJ - 1;
    IF SUBJ = MIN THEN
    BEGIN
        FLAG := 1;
        MIN := ADDJ+1;
    END
    ELSE
        FLAG := 2;
        MID := SUBJ;
    END;
END; { END CASE }
SEEK(FILEGR12,MID-1);
READ(FILEGR12,RECGR12);
SUM_QSAM := SUM_QSAM + RECGR12.GROUP12;
II := II + 1;
END;
CLOSE(FILEGR12);
END;
END;      { END CASE }
END;
[*****END RAN_SAM*****]
PROCEDURE QUOTA_MIN(NO_SAM,NOS,NN : INTEGER);
VAR
    II : INTEGER;
BEGIN
    IF NO_SAM = 0 THEN
        EXIT;
    CASE NN OF
        1:
            BEGIN
                ASSIGN(FILEGR1,'G1D01');
                RESET(FILEGR1);
                FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
                BEGIN
                    READ(FILEGR1,RECGR1);
                    SUM_QSAM := SUM_QSAM + RECGR1.GROUP1;
                END;
                CLOSE(FILEGR1);
            END;
        2:
            BEGIN
                ASSIGN(FILEGR2,'G1D02');
                RESET(FILEGR2);
                FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
                BEGIN
                    READ(FILEGR2,RECGR2);
                    SUM_QSAM := SUM_QSAM + RECGR2.GROUP2;
                END;
                CLOSE(FILEGR2);
            END;
        3:
            BEGIN

```

```

ASSIGN(FILEGR3,'G1D03');
RESET(FILEGR3);
FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
BEGIN
    READ(FILEGR3,RECGR3);
    SUM_QSAM := SUM_QSAM + RECGR3.GROUP3;
END;
CLOSE(FILEGR3);
END;
4:
BEGIN
    ASSIGN(FILEGR4,'G1D04');
    RESET(FILEGR4);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        READ(FILEGR4,RECGR4);
        SUM_QSAM := SUM_QSAM + RECGR4.GROUP4;
    END;
    CLOSE(FILEGR4);
END;
5:
BEGIN
    ASSIGN(FILEGR5,'G1D05');
    RESET(FILEGR5);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        READ(FILEGR5,RECGR5);
        SUM_QSAM := SUM_QSAM + RECGR5.GROUP5;
    END;
    CLOSE(FILEGR5);
END;
6:
BEGIN
    ASSIGN(FILEGR6,'G1D06');
    RESET(FILEGR6);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        READ(FILEGR6,RECGR6);
        SUM_QSAM := SUM_QSAM + RECGR6.GROUP6;
    END;
    CLOSE(FILEGR6);
END;
7:
BEGIN
    ASSIGN(FILEGR7,'G1D07');
    RESET(FILEGR7);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        READ(FILEGR7,RECGR7);
        SUM_QSAM := SUM_QSAM + RECGR7.GROUP7;
    END;
    CLOSE(FILEGR7);
END;
8:
BEGIN

```

```

ASSIGN(FILEGR8,'G1D08');RESET(FILEGR8);
FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
BEGIN
    READ(FILEGR8,RECGR8);
    SUM_QSAM := SUM_QSAM + RECGR8.GROUP8;
END;
CLOSE(FILEGR8);
END;
9:
BEGIN
    ASSIGN(FILEGR9,'G1D09');
    RESET(FILEGR9);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        READ(FILEGR9,RECGR9);
        SUM_QSAM := SUM_QSAM + RECGR9.GROUP9;
    END;
    CLOSE(FILEGR9);
END;
10:
BEGIN
    ASSIGN(FILEGR10,'G1D010');
    RESET(FILEGR10);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        READ(FILEGR10,RECGR10);
        SUM_QSAM := SUM_QSAM + RECGR10.GROUP10;
    END;
    CLOSE(FILEGR10);
END;
11:
BEGIN
    ASSIGN(FILEGR11,'G1D011');
    RESET(FILEGR11);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        READ(FILEGR11,RECGR11);
        SUM_QSAM := SUM_QSAM + RECGR11.GROUP11;
    END;
    CLOSE(FILEGR11);
END;
12:
BEGIN
    ASSIGN(FILEGR12,'G1D012');
    RESET(FILEGR12);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        READ(FILEGR12,RECGR12);
        SUM_QSAM := SUM_QSAM + RECGR12.GROUP12;
    END;
    CLOSE(FILEGR12);
END;
END; { END CASE}
END;
[***** END QUOTA MIN *****]

```

```

PROCEDURE QUOTA_MAX(NO_SAM,NOS,NN : INTEGER);
VAR
  III,II : INTEGER;

BEGIN
  IF NO_SAM = 0 THEN
    EXIT;
  III := NOS;
  CASE NN OF
    1:
      BEGIN
        ASSIGN(FILEGR1,'G1D01');
        RESET(FILEGR1);
        FOR II := 1 TO NO_SAM DO
          BEGIN
            III := III - 1;
            SEEK(FILEGR1,III);
            READ(FILEGR1,RECGR1);
            SUM_QSAM := SUM_QSAM + RECGR1.GROUP1;
          END;
        CLOSE(FILEGR1);
      END;
    2:
      BEGIN
        ASSIGN(FILEGR2,'G1D02');
        RESET(FILEGR2);
        FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
          BEGIN
            III := III - 1;
            SEEK(FILEGR2,III);
            READ(FILEGR2,RECGR2);
            SUM_QSAM := SUM_QSAM + RECGR2.GROUP2;
          END;
        CLOSE(FILEGR2);
      END;
    3:
      BEGIN
        ASSIGN(FILEGR3,'G1D03');
        RESET(FILEGR3);
        FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
          BEGIN
            III := III - 1;
            SEEK(FILEGR3,III);
            READ(FILEGR3,RECGR3);
            SUM_QSAM := SUM_QSAM + RECGR3.GROUP3;
          END;
        CLOSE(FILEGR3);
      END;
    4:
      BEGIN
        ASSIGN(FILEGR4,'G1D04');
        RESET(FILEGR4);
        FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
          BEGIN
            III := III - 1;

```



```
        SEEK(FILEGR4,III);
        READ(FILEGR4,RECGR4);
        SUM_QSAM := SUM_QSAM + RECGR4.GROUP4;
    END;
    CLOSE(FILEGR4);
END;
5:
BEGIN
    ASSIGN(FILEGR5,'G1D05');
    RESET(FILEGR5);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        III := III - 1;
        SEEK(FILEGR5,III);
        READ(FILEGR5,RECGR5);
        SUM_QSAM := SUM_QSAM + RECGR5.GROUP5;
    END;
    CLOSE(FILEGR5);
END;
6:
BEGIN
    ASSIGN(FILEGR6,'G1D06');
    RESET(FILEGR6);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        III := III - 1;
        SEEK(FILEGR6,III);
        READ(FILEGR6,RECGR6);
        SUM_QSAM := SUM_QSAM + RECGR6.GROUP6;
    END;
    CLOSE(FILEGR6);
END;
7:
BEGIN
    ASSIGN(FILEGR7,'G1D07');
    RESET(FILEGR7);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        III := III - 1;
        SEEK(FILEGR7,III);
        READ(FILEGR7,RECGR7);
        SUM_QSAM := SUM_QSAM + RECGR7.GROUP7;
    END;
    CLOSE(FILEGR7);
END;
8:
BEGIN
    ASSIGN(FILEGR8,'G1D08');
    RESET(FILEGR8);
    FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
    BEGIN
        III := III - 1;
        SEEK(FILEGR8,III);
        READ(FILEGR8,RECGR8);
        SUM_QSAM := SUM_QSAM + RECGR8.GROUP8;
```

```

        END;
        CLOSE(FILEGR8);
    END;
9:
    BEGIN
        ASSIGN(FILEGR9,'G1D09');
        RESET(FILEGR9);
        FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
        BEGIN
            III := III - 1;
            SEEK(FILEGR9,III);
            READ(FILEGR9,RECGR9);
            SUM_QSAM := SUM_QSAM + RECGR9.GROUP9;
        END;
        CLOSE(FILEGR9);
    END;
10:
    BEGIN
        ASSIGN(FILEGR10,'G1D010');
        RESET(FILEGR10);
        FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
        BEGIN
            III := III - 1;
            SEEK(FILEGR10,III);
            READ(FILEGR10,RECGR10);
            SUM_QSAM := SUM_QSAM + RECGR10.GROUP10;
        END;
        CLOSE(FILEGR10);
    END;
11:
    BEGIN
        ASSIGN(FILEGR11,'G1D011');
        RESET(FILEGR11);
        FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
        BEGIN
            III := III - 1;
            SEEK(FILEGR11,III);
            READ(FILEGR11,RECGR11);
            SUM_QSAM := SUM_QSAM + RECGR11.GROUP11;
        END;
        CLOSE(FILEGR11);
    END;
12:
    BEGIN
        ASSIGN(FILEGR12,'G1D012');
        RESET(FILEGR12);
        FOR II := 1 TO NO_SAM DO { RANDOM SAMPLE }
        BEGIN
            III := III - 1;
            SEEK(FILEGR12,III);
            READ(FILEGR12,RECGR12);
            SUM_QSAM := SUM_QSAM + RECGR12.GROUP12;
        END;
        CLOSE(FILEGR12);
    END;
END;

```

```

        END; { END CASE}
    END;

BEGIN
    SUM_QSAM := 0.0;
    IF S = 1 THEN
        FOR NN := 1 TO NOG DO
            QUOTA_MIN(NO_SAM[NN],NO[NN],NN)
        ELSE
            IF S = 2 THEN
                FOR NN := 1 TO NOG DO
                    QUOTA_MAX(NO_SAM[NN],NO[NN],NN)
                ELSE
                    FOR NN := 1 TO NOG DO
                        QUOTA_SAM(NO_SAM[NN],NO[NN],NN);
                    MEAN_QUOTA1 := SUM_QSAM / I_SAM[I];
                    ERROR_QUOTA1 := ABS(MEAN_QUOTA1 - I_MEAN_NORM) * 100 / I_MEAN_NORM;
                    IF (S = 1) OR (S = 2) THEN
                        BEGIN
                            WRITELN(LST);
                            WRITELN(LST, ' ',MEAN_QUOTA1:7:4,' ',ERROR_QUOTA1:5:4);
                        END
                    ELSE
                        BEGIN
                            MEAN_QUOTA := MEAN_QUOTA + MEAN_QUOTA1;
                            ERROR_QUOTA := ERROR_QUOTA + ERROR_QUOTA1;
                        END;
                    END;
                END;
            END;
        END;
    END;
    [*****END QUOTA_SAM*****]

PROCEDURE SAM_LEVEL(G:INTEGER);
VAR
    S,NO_TMP,JJ,J : INTEGER;
    SUM_GROUP,SUM_GROUP2,MEAN_GROUP,STD_GROUP,STD_GROUP2 :REAL;

    PROCEDURE CALCV(NN:INTEGER);
    VAR CVG : REAL;
    BEGIN
        IF I = 1 THEN
            BEGIN
                MEAN_GROUP := SUM_GROUP / NO[NN];
                STD_GROUP := (SUM_GROUP2 - (NO[NN] * SQR(MEAN_GROUP))) / (NO[NN] - 1);
                STD_GROUP2 := SQR(STD_GROUP);
                CVG := (STD_GROUP2 / MEAN_GROUP) * 100;
                WRITELN(LST,'POP STRATIFIED GROUP ',NN,' MEAN = ',MEAN_GROUP:8:2,
                    ',STD = ',STD_GROUP2:5:2,' C.V. = ',CVG:3:2);
            END;
        END;
    END;

PROCEDURE CALSUM;
BEGIN
    IF I = 1 THEN
        BEGIN
            SUM_GROUP := SUM_GROUP + RECX.Y;

```

```

        SUM_GROUP2 := SUM_GROUP2 + SQR(RECX.Y);
    END;
END;

BEGIN
    MEAN_STRAT := 0.0;
    MEAN_QUOTA := 0.0;
    ERROR_STRAT := 0.0;
    ERROR_QUOTA := 0.0;
    FOR S := 1 TO 12 DO
        NO[S] := 0;
    J := 1;
    JJ := 1;
    SUM_GROUP := 0.0;
    SUM_GROUP2 := 0.0;
    ASSIGN (FILEXY, 'XY1D01');
    RESET (FILEXY);
    WHILE (X[JJ] = 1.0) DO
    BEGIN
        NO[1] := NO[1] + 1;
        READ(FILEXY, RECX);
        JJ := JJ + 1;
        YG[J] := RECX.Y;
        J := J + 1;
        CALSUM;
    END;
    quicksort(YG, 1, NO[1]);
    ASSIGN(FILEGR1, 'G1D01');
    REWRITE(FILEGR1);
    FOR J := 1 TO NO[1] DO
    BEGIN
        RECGR1.GROUP1 := YG[J];
        WRITE(FILEGR1, RECGR1);
    END;
    CLOSE(FILEGR1);
    CALCV(1);
    SUM_GROUP := 0.0;
    SUM_GROUP2 := 0.0;
    J := 1;
    WHILE (X[JJ] = 2.0) DO
    BEGIN
        READ(FILEXY, RECX);
        NO[2] := NO[2] + 1;
        JJ := JJ + 1;
        YG[J] := RECX.Y;
        J := J + 1;
        CALSUM;
    END;
    quicksort(YG, 1, NO[2]);
    ASSIGN(FILEGR2, 'G1D02');
    REWRITE(FILEGR2);
    FOR J := 1 TO NO[2] DO
    BEGIN
        RECGR2.GROUP2 := YG[J];
        WRITE(FILEGR2, RECGR2);
    END;
END;

```

```

END;
CLOSE(FILEGR2);
CALCV(2);
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
J := 1;
WHILE (X[JJ] = 3.0) DO
BEGIN
  READ(FILEXY,RECY);
  NO[3] := NO[3] + 1;
  JJ := JJ + 1;
  YG[J] := RECY.Y;
  J := J + 1;
  CALSUM;
END;
quicksort(YG,1,NO[3]);
ASSIGN(FILEGR3,'G1D03');
REWRITE(FILEGR3);
FOR J := 1 TO NO[3] DO
BEGIN
  RECGR3.GROUP3 := YG[J];
  WRITE(FILEGR3,RECGR3);
END;
CLOSE(FILEGR3);
CALCV(3);
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
IF (G = 3) OR (G = 2) OR (G = 1) THEN
BEGIN
  J := 1;
  WHILE (X[JJ] = 4.0) DO
  BEGIN
    READ(FILEXY,RECY);
    NO[4] := NO[4] + 1;
    JJ := JJ + 1;
    YG[J] := RECY.Y;
    J := J + 1;
    CALSUM;
  END;
  quicksort(YG,1,NO[4]);
  ASSIGN(FILEGR4,'G1D04');
  REWRITE(FILEGR4);
  FOR J := 1 TO NO[4] DO
  BEGIN
    RECGR4.GROUP4 := YG[J];
    WRITE(FILEGR4,RECGR4);
  END;
  CLOSE(FILEGR4);
  CALCV(4);
  SUM_GROUP := 0.0;
  SUM_GROUP2 := 0.0;
  J := 1;
  WHILE (X[JJ] = 5.0) DO
  BEGIN
    READ(FILEXY,RECY);

```

```

NO[5] := NO[5] + 1;
JJ := JJ + 1;
YG[J] := RECX.Y;
J := J + 1;
CALSUM;
END;
quicksort(YG,1,NO[5]);
ASSIGN(FILEGR5,'G1D05');
REWRITE(FILEGR5);
FOR J := 1 TO NO[5] DO
BEGIN
    RECGR5.GROUP5 := YG[J];
    WRITE(FILEGR5,RECGR5);
END;
CLOSE(FILEGR5);
CALCV(5);
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
J := 1;
WHILE (X[JJ] = 6.0) DO
BEGIN
    READ(FILEXY,RECX);
    NO[6] := NO[6] + 1;
    JJ := JJ + 1;
    YG[J] := RECX.Y;
    J := J + 1;
    CALSUM;
END;
quicksort(YG,1,NO[6]);
ASSIGN(FILEGR6,'G1D06');
REWRITE(FILEGR6);
FOR J := 1 TO NO[6] DO
BEGIN
    RECGR6.GROUP6 := YG[J];
    WRITE(FILEGR6,RECGR6);
END;
CLOSE(FILEGR6);
CALCV(6);
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
IF (G = 2) OR (G = 1) THEN
BEGIN
    J := 1;
    WHILE (X[JJ] = 7.0) DO
    BEGIN
        READ(FILEXY,RECX);
        NO[7] := NO[7] + 1;
        JJ := JJ + 1;
        YG[J] := RECX.Y;
        J := J + 1;
        CALSUM;
    END;
    quicksort(YG,1,NO[7]);
    ASSIGN(FILEGR7,'G1D07');
    REWRITE(FILEGR7);

```

```

FOR J := 1 TO NO[7] DO
BEGIN
    RECGR7.GROUP7 := YG[J];
    WRITE(FILEGR7,RECGR7);
END;
CLOSE(FILEGR7);
CALCV(7);
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
J := 1;
WHILE (X[JJ] = 8.0) DO
BEGIN
    READ(FILEXY,RECY);
    NO[8] := NO[8] + 1;
    JJ := JJ + 1;
    YG[J] := RECY.Y;
    J := J + 1;
    CALSUM;
END;
quicksort(YG,1,NO[8]);
ASSIGN(FILEGR8,'G1D08');
REWRITE(FILEGR8);
FOR J := 1 TO NO[8] DO
BEGIN
    RECGR8.GROUP8 := YG[J];
    WRITE(FILEGR8,RECGR8);
END;
CLOSE(FILEGR8);
CALCV(8);
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
J := 1;
WHILE (X[JJ] = 9.0) DO
BEGIN
    READ(FILEXY,RECY);
    NO[9] := NO[9] + 1;
    JJ := JJ + 1;
    YG[J] := RECY.Y;
    J := J + 1;
    CALSUM;
END;
quicksort(YG,1,NO[9]);
ASSIGN(FILEGR9,'G1D09');
REWRITE(FILEGR9);
FOR J := 1 TO NO[9] DO
BEGIN
    RECGR9.GROUP9 := YG[J];
    WRITE(FILEGR9,RECGR9);
END;
CLOSE(FILEGR9);
CALCV(9);
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
IF G = 1 THEN
BEGIN

```

```

J := 1;
WHILE (X[JJ] = 10.0) DO
BEGIN
  READ(FILEXY,RECY);
  NO[10] := NO[10] + 1;
  JJ := JJ + 1;
  YG[J] := RECY.Y;
  J := J + 1;
  CALSUM;
END;
quicksort(YG,1,NO[10]);
ASSIGN(FILEGR10,'G1D010');
REWRITE(FILEGR10);
FOR J := 1 TO NO[10] DO
BEGIN
  RECGR10.GROUP10 := YG[J];
  WRITE(FILEGR10,RECGR10);
END;
CLOSE(FILEGR10);
CALCV(10);
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
J := 1;
WHILE (X[JJ] = 11.0) DO
BEGIN
  READ(FILEXY,RECY);
  NO[11] := NO[11] + 1;
  JJ := JJ + 1;
  YG[J] := RECY.Y;
  J := J + 1;
  CALSUM;
END;
quicksort(YG,1,NO[11]);
ASSIGN(FILEGR11,'G1D011');
REWRITE(FILEGR11);
FOR J := 1 TO NO[11] DO
BEGIN
  RECGR11.GROUP11 := YG[J];
  WRITE(FILEGR11,RECGR11);
END;
CLOSE(FILEGR11);
CALCV(11);
SUM_GROUP := 0.0;
SUM_GROUP2 := 0.0;
J := 1;
WHILE NOT EOF(FILEXY) AND (X[JJ] = 12.0) DO
BEGIN
  READ(FILEXY,RECY);
  NO[12] := NO[12] + 1;
  JJ := JJ + 1;
  YG[J] := RECY.Y;
  J := J + 1;
  CALSUM;
END;
quicksort(YG,1,NO[12]);

```



```

ASSIGN(FILEGR12,'G1D012');
REWRITE(FILEGR12);
FOR J := 1 TO NO[12] DO
BEGIN
    RECGR12.GROUP12 := YG[J];
    WRITE(FILEGR12,RECGR12);
END;
CLOSE(FILEGR12);
CALCV(12);
CLOSE(FILEXY);
NO_TMP := 0;
FOR J := 1 TO 11 DO
BEGIN
    NO_SAM[J] := ROUND((NO[J] / I_COUNT)*I_SAM[I] );
    NO_TMP := NO_TMP + NO_SAM[J];
END;
NO_SAM[12]:= I_SAM[I] - NO_TMP;
Writeln(LST);
Writeln(LST,'MEAN STRAT WHEN X = 12 LEVEL : ':40);
FOR S := 1 TO S_COUNT DO
    RAN_STRAT(12,S);
Writeln(LST,'MEAN QUOTA WHEN X = 12 LEVEL : ':40);
FOR S := 1 TO S_COUNT DO
    RAN_QUOTA(12,S);
END
ELSE
BEGIN
    CLOSE(FILEXY);
    NO_TMP := 0;
    FOR J := 1 TO 8 DO
    BEGIN
        NO_SAM[J] := ROUND((NO[J] / I_COUNT)*I_SAM[I] );
        NO_TMP := NO_TMP + NO_SAM[J];
    END;
    NO_SAM[9]:= I_SAM[I] - NO_TMP;
    Writeln('MEAN STRAT WHEN X = 9 LEVEL : ':40);}
    Writeln(LST);
    Writeln(LST,'MEAN STRAT WHEN X = 9 LEVEL : ':40);
    FOR S := 1 TO S_COUNT DO
        RAN_STRAT(9,S);
    Writeln('MEAN QUOTA WHEN X = 9 LEVEL : ':40);}
    Writeln(LST,'MEAN QUOTA WHEN X = 9 LEVEL : ':40);
    FOR S := 1 TO S_COUNT DO
        RAN_QUOTA(9,S);
    END
END
ELSE
BEGIN
    CLOSE(FILEXY);
    NO_TMP := 0;
    FOR J := 1 TO 5 DO
    BEGIN
        NO_SAM[J] := ROUND((NO[J] / I_COUNT)*I_SAM[I] );
        NO_TMP := NO_TMP + NO_SAM[J];
    END;

```

```

        NO_SAM[6]:= I_SAM[I] - NO_TMP;
        WRITELN('MEAN STRAT WHEN X = 6 LEVEL : ':40);}
        WRITELN(LST);
        WRITELN(LST,'MEAN STRAT WHEN X = 6 LEVEL : ':40);
        FOR S := 1 TO S_COUNT DO
            RAN_STRAT(6,S);
        {
            WRITELN('MEAN QUOTA WHEN X = 6 LEVEL : ':40);}
        WRITELN(LST,'MEAN QUOTA WHEN X = 6 LEVEL : ':40);
        FOR S := 1 TO S_COUNT DO
            RAN_QUOTA(6,S);
    END
END
ELSE
BEGIN
    CLOSE(FILEXY);
    NO_TMP := 0;
    FOR J := 1 TO 2 DO
        BEGIN
            NO_SAM[J] := ROUND((NO[J] / I_COUNT)*I_SAM[I] );
            NO_TMP := NO_TMP + NO_SAM[J];
        END;
    NO_SAM[3]:= I_SAM[I] - NO_TMP;
    {
        WRITELN('MEAN STRAT WHEN X = 3 LEVEL : ':40);}
    WRITELN(LST);
    WRITELN(LST,'MEAN STRAT WHEN X = 3 LEVEL : ':40);
    FOR S := 1 TO S_COUNT DO
        RAN_STRAT(3,S);
    {
        WRITELN('MEAN QUOTA WHEN X = 3 LEVEL : ':40);}
    WRITELN(LST,'MEAN QUOTA WHEN X = 3 LEVEL : ':40);
    FOR S := 1 TO S_COUNT DO
        RAN_QUOTA(3,S);
    END;

    { *** END CREATE GROUP *** }
    MEAN_STRAT := MEAN_STRAT / S_COUNT;
    ERROR_STRAT := ERROR_STRAT / S_COUNT;
    MEAN_QUOTA := MEAN_QUOTA / (S_COUNT - 2);
    ERROR_QUOTA := ERROR_QUOTA / (S_COUNT - 2);
    WRITE(LST,'MEAN_STRAT ',G,' = ',MEAN_STRAT:9:4);
    WRITE(LST,' ERROR_STRAT ',G,' = ',ERROR_STRAT:5:4);
    WRITELN(LST);
    WRITE(LST,'MEAN_QUOTA ',G,' = ',MEAN_QUOTA:9:4);
    WRITELN(LST,' ERROR_QUOTA ',G,' = ',ERROR_QUOTA:5:4);
END;

BEGIN {MAIN}
    IX := 65479; {65539} {7257371} {9753451} {START VALUE PROGRAM}
    {
        IX := 90147;}{55555}{926}{81543} {90367}{63087}{51483}{65631}
    IX1 := 12345;
    IX2 := 12345;
    KK := 0;
    CLRSCR;
    I_STD_NORM [1] := 200;
    CORR_X_Y[1] := 0.1;
    I_STD_NORM [2] := 200;

```

```

CORR_X_Y[2]      := 0.5;
I_STD_NORM[3]    := 200;
CORR_X_Y[3]      := 0.9;
I_STD_NORM[4]    := 500;
CORR_X_Y[4]      := 0.1;
I_STD_NORM[5]    := 500;
CORR_X_Y[5]      := 0.5;
I_STD_NORM[6]    := 500;
CORR_X_Y[6]      := 0.9;
I_STD_NORM[7]    := 1000;
CORR_X_Y[7]      := 0.1;
I_STD_NORM[8]    := 1000;
CORR_X_Y[8]      := 0.5;
I_STD_NORM[9]    := 1000;
CORR_X_Y[9]      := 0.9;
I_STD_NORM[10]   := 2000;
CORR_X_Y[10]     := 0.1;
I_STD_NORM[11]   := 2000;
CORR_X_Y[11]     := 0.5;
I_STD_NORM[12]   := 2000;
CORR_X_Y[12]     := 0.9;
I_SAM[1]         := 250;
I_SAM[2]         := 500;
I_SAM[3]         := 750;
FOR K:= 1 TO K_COUNT DO                                (** REPEAT SIMULATION **)
BEGIN
  CVP := (I_STD_NORM[K] / I_MEAN_NORM ) * 100;
  WRITELN(LST);
  WRITELN(LST);
  WRITELN(LST, '***** POPULATION ',K, ' *****');
  WRITE(LST, 'I_COUNT = ',I_COUNT:4);
  WRITE(LST, ' ', 'TOTAL REPEAT = ',K_COUNT:4);
  WRITE(LST, ' ', 'TOTAL RANDOM = ',S_COUNT:4);
  WRITELN(LST);
  WRITE(LST, 'MEAN = ',I_MEAN_NORM:5:2);
  WRITE(LST, ' ', 'STD = ',I_STD_NORM[K]:5:2);
  WRITE(LST, ' ', 'C.V. = ',CVP:3:2);
  WRITELN(LST);
  WRITELN(LST, 'CORR_X_Y = ',CORR_X_Y[K]:4:2);
  SIM_NORM;
  PRTPDATA;
  FOR I := 1 TO 3 DO
  BEGIN
    WRITELN(LST);
    WRITELN(LST, '**** I_SAM = ',I_SAM[I]:4, ' ****');
    FOR G := 1 TO 4 DO
    BEGIN
      TRANX(G);
      SAM_LEVEL(G);                                (** SAMPLING FOR STRATIFIED & QUOTA **)
    END;
    PRTPG(G);
  END;
  IX := IX + K;
END; { END LOOP K := 1 TO K_COUNT }
END.

```

ประวัติผู้เขียน

ชื่อ นายประสา สุวัฒนพันธุ์กุล เกิดวันที่ 11 ตุลาคม พ.ศ. 2503 ที่กรุงเทพฯ
 เรียนจบชั้นประถมศึกษาที่โรงเรียนหงส์สุวรรณังก์ จบมัธยมศึกษาที่โรงเรียนวัดบวรนิเวศ และจบ
 ปริญญาตรีคณะศิลปศาสตร์ เอกทางด้านสถิติ เมื่อปีการศึกษา 2526 เข้าศึกษาต่อระดับปริญญา
 มหาคบัณฑิตในภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2530 ปัจจุบัน
 ทำงานที่ธนาคารกสิกรไทย สำนักงานใหญ่ ฝ่ายคอมพิวเตอร์



ศูนย์วิทยทรัพยากร
 จุฬาลงกรณ์มหาวิทยาลัย