

รายการอ้างอิง

ภาษาไทย

- ดาว คงศิริวัฒนา “การเปรียบเทียบอำนาจการทดสอบของตัวสถิติสำหรับการทดสอบความคลาดเคลื่อนที่มีความแปรปรวนไม่คงที่ในการวิเคราะห์ความถดถอยเชิงเส้น” วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2532
- ธีระพร วีระถาวร “ความน่าจะเป็นกับการประยุกต์” กรุงเทพมหานคร ภาควิชาสถิติ จุฬาลงกรณ์มหาวิทยาลัย, 2537
- วินัย โพธิ์สุวรรณ “การเปรียบเทียบตัวสถิติสำหรับการทดสอบความเท่ากันของความแปรปรวน” วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2533

ภาษาอังกฤษ

- Damodar N.Gujarati, Basic Econometrics. 2nd edition, New York: McGraw-Hill.
- Edward J.Dudewiez and Satya N.mishra, Modern Mathematical Statistics ., New York: John Wiley and Son.
- Subhash C.Sharma and Carmelo Giaccotto, “Power and robustness of Jackknife and Likelihood Ratio tests for Grouped Heteroscedasticity.” *Econometrics*, vol.4, no.3, 1991:343-372.

ภาคผนวก

{SN+}

Program Test;

(* uses Btrtools; *)

const

(* three value change for use n =40 ,80 ,120 *)

MAXNUM1= 120; (* number of sample : 40 or 80 or 120 *)

MAXNUM2= 60; (* number of random : 20 or 40 or 60 *)

IX = 30; (* number of inner loop : 10 or 20 or 30 *)

VII = 57/2; (* [(n-k)/wi-1], wi=n/ni *)

VI2 = 113/4; (* [(n-k-1)/wi-1], wi=n/ni *)

P1 = 0.10; (* percent of scale contaminate *)

C1 = 10; (* scale contaminate *)

PI = 3.1415926;

B0 = 1; (* parameter of regression *)

B1 = 1; (* parameter of regression *)

F_1_3_116 = 3.962;

F_5_3_116 = 2.685;

X_1_3 = 11.3;

X_5_3 = 7.81;

type (* for write file *)

File_Name . = Record

Value : packed array [1..14] of char;

end;

Position = Record

Blk : packed array [1..128] of Byte;

end;

var

RM : array [1..4] of real;

RV : array [1..4] of real;

RX : array [1..MAXNUM1] of real;

EX : array [1..MAXNUM1] of real;

EX2 : array [1..MAXNUM1] of real;

EXY : array [1..MAXNUM1] of real;
 ES : array [1..MAXNUM1] of real;
 ER : array [1..MAXNUM1] of real;
 em : array [1..maxnum1] of real;
 EY : array [1..MAXNUM1] of real;
 EY2 : array [1..MAXNUM1] of real;
 EH : array [1..MAXNUM1] of real;
 EYH : array [1..MAXNUM1] of real;

SEX : array [1..4] of real;
 SEX2 : array [1..4] of real;
 SEXY : array [1..4] of real;
 SEY : array [1..4] of real;
 SEY2 : array [1..4] of real;
 BH0 : array [1..4] of real;
 BH1 : array [1..4] of real;
 i : integer;
 j : integer;
 ZONE : real;
 ZTWO : real;

SE : array [1..4] of real;
 SSE : array [1..4] of real;
 SI2 : array [1..4] of real;
 SSI : array [1..4] of real;

ASE : array [1..MAXNUM1] of real;
 SE1 : array [1..MAXNUM1] of real;
 ASE12 : array [1..MAXNUM1] of real;
 SIJ2 : array [1..MAXNUM1] of real;
 UIJ : array [1..MAXNUM1] of real;
 SUIJ : array [1..4] of real;
 UI : array [1..4] of real;

```
BHAT1 : real;
BHAT0 : real;
SXY   : real;
SX    : real;
SY    : real;
SX2   : real;

BHML0 : real;
BHML1 : real;
SSEH  : array [1..4] of real;
SSEHT : array [1..4] of real;
EHML  : array [1..MAXNUM1] of real;
EYHML : array [1..MAXNUM1] of real;
SLML2 : real;
temp1  : real;
temp2  : real;
temp3  : real;
temp4  : real;
temp5  : real;
temp6  : real;
temp7  : real;
temp8  : real;
temp9  : real;

SUIJT : real;
UT     : real;
EUI    : real;
EUIJ  : array [1..4] of real;
SEUIJ : real;
JACK1  : real;
SL2    : real;
LROLS  : real;
SUBB   : real;
SUBL   : real;
```

```

BART : real;
Zrng : array [1..100] of longint;
Zget : longint;
usewait: char;
Found_0: boolean;
Acc_J_1,Rej_J_1,Acc_L_1,Rej_L_1,Acc_B_1,Rej_B_1 :integer;
Acc_J_5,Rej_J_5,Acc_L_5,Rej_L_5,Acc_B_5,Rej_B_5 :integer;
BtrStatus : integer;
Cnt_Loop : integer;
Cnt_exp : integer;
Cnt_Seq1 : integer;
Cnt_Seq2 : integer;
Cnt_Seq3 : integer;
Seed : integer;
J_1,J_5,L_1,L_5,B_1,B_5 : integer;

```

```

procedure Randdf;

```

```

begin ( RANDDF )

```

```

( Set the seeds for all 100 streams )

```

```

Zrng[01] := 1973272912;Zrng[02] := 281629770;Zrng[03] := 20006270;Zrng[04] :=1280689831;
Zrng[05] := 2096730329;Zrng[06] :=1933576050;Zrng[07] := 913566091;Zrng[08] := 246780520;
Zrng[09] := 1363774876;Zrng[10] := 604901985;Zrng[11] :=1511192140;Zrng[12] :=1259851944;
Zrng[13] := 824064364;Zrng[14] := 150493284;Zrng[15] := 242708531;Zrng[16] := 75253171;
Zrng[17] := 1964472944;Zrng[18] :=1202299975;Zrng[19] := 233217322;Zrng[20] :=1911216000;
Zrng[21] := 726370533;Zrng[22] := 403498145;Zrng[23] := 993232223;Zrng[24] :=1103205531;
Zrng[25] := 762430696;Zrng[26] :=1922803170;Zrng[27] :=1385516923;Zrng[28] := 76271663;
Zrng[29] := 413682397;Zrng[30] := 726466604;Zrng[31] := 336157058;Zrng[32] :=1432650381;
Zrng[33] := 1120463904;Zrng[34] := 595778810;Zrng[35] := 877722890;Zrng[36] :=1046574445;
Zrng[37] := 68911991;Zrng[38] :=2088367019;Zrng[39] := 748545416;Zrng[40] := 622401386;
Zrng[41] := 2122378830;Zrng[42] := 640690903;Zrng[43] :=1774806513;Zrng[44] :=2132545692;
Zrng[45] := 2079249579;Zrng[46] := 78130110;Zrng[47] := 852776735;Zrng[48] :=1187867272;
Zrng[49] := 1351423507;Zrng[50] :=1645973084;Zrng[51] :=1997049139;Zrng[52] := 922510944;
Zrng[53] := 2045512870;Zrng[54] := 898585771;Zrng[55] := 243649545;Zrng[56] :=1004818771;
Zrng[57] := 773686062;Zrng[58] := 403188473;Zrng[59] := 372279877;Zrng[60] :=1901633463;

```

```

Zrng[61] := 498067494;Zrng[62] :=2087759558;Zrng[63] := 493157915;Zrng[64] := 597104727;
Zrng[65] := 1530940798;Zrng[66] :=1814496276;Zrng[67] := 536444882;Zrng[68] :=1663153658;
Zrng[69] := 855503735;Zrng[70] := 67784357;Zrng[71] :=1432404475;Zrng[72] := 619691088;
Zrng[73] := 119025595;Zrng[74] := 880802310;Zrng[75] := 176192644;Zrng[76] :=1116780070;
Zrng[77] := 277854671;Zrng[78] :=1366580350;Zrng[79] :=1142483975;Zrng[80] :=2026948561;
Zrng[81] := 1053920743;Zrng[82] := 786262391;Zrng[83] :=1792203830;Zrng[84] :=1494667770;
Zrng[85] := 1923011392;Zrng[86] :=1433700034;Zrng[87] :=1244184613;Zrng[88] :=1147297105;
Zrng[89] := 539712780;Zrng[90] :=1545929719;Zrng[91] := 190641742;Zrng[92] :=1645390429;
Zrng[93] := 264907697;Zrng[94] := 620389253;Zrng[95] :=1502074852;Zrng[96] := 927711160;
Zrng[97] := 364849192;Zrng[98] :=2049576050;Zrng[99] := 638580085;Zrng[100]:= 547070247;
end; | RANDDF |

```

```
function Rand(Stream : integer): real; {Generate the next random number}
```

```

const
  B2E15 = 32768;
  B2E16 = 65536;
  Modlus = 2147483647;
  Mult1 = 24112;
  Mult2 = 26143;

var
  Hi15,Hi31,Low15,Lowprd,Ovflow,Zi : longint;

begin
  Zi := Zrng[Stream];
  Hi15 := Zi DIV B2E16;
  Lowprd := (Zi-Hi15*B2E16)*Mult1;
  Low15 := Lowprd DIV B2E16;
  Hi31 := Hi15*Mult1+Low15;
  Ovflow := Hi31 DIV B2E15;
  Zi := (((Lowprd-Low15*B2E16)-Modlus)+(Hi31-Ovflow*B2E15)*B2E16)+Ovflow;
  if Zi <0 then begin
    Zi := Zi+Modlus;
  end; {endif}
  Hi15 := Zi DIV B2E16;
  Lowprd := (Zi-Hi15*B2E16)*Mult2;

```

```

Low15 := Lowprd DIV B2E16;
Hi31  := Hi15*Mult2+Low15;
Ovflow := Hi31 DIV B2E15;
Zi     := (((Lowprd-Low15*B2E16)-Modlus)+(Hi31-Ovflow*B2E15)*B2E16)+Ovflow;
if Zi < 0 then begin
  Zi := Zi+Modlus;
end; {endif}
Zrng[Stream] := Zi;
Rand := (2*(Zi DIV 265) + 1) / 16777216.0;
end;

procedure Randst(Zset :longint;Stream : integer);
begin { RANDST }
  Zrng[Stream] :=Zset;
end; { RANDST }

function Randgt(Stream : integer):longint;
begin { RANDGT }
  Randgt := Zrng[Stream];
end; { RANDGT }

procedure InitialVar;
var
  i :integer;
begin { }
  (* Seed := 0;
  randomize;
  Randdf:*)
  (* define_ex:*)
  for i := 1 to 4 do begin
    SEX[i] := 0.00;
    SEX2[i] := 0.00;
    SEXY[i] := 0.00;
    SEY[i] := 0.00;
    SEY2[i] := 0.00;
  end;
end;

```



```

BH0[i] := 0.00;
BH1[i] := 0.00;
BHAT0 := 0.00;
BHAT1 := 0.00;
SXY := 0.00;
SX := 0.00;
SY := 0.00;
SX2 := 0.00;

SE[i] := 0.00;
SSE[i] := 0.00;
SI2[i] := 0.00;
SSI[i] := 0.00;
SUIJ[i] := 0.00;
UI[i] := 0.00;
EUIJ[i] := 0.00;

SSEH[i] := 0.00;
SSEHT[i] := 0.00;
SLML2 := 0.00;
EHML[i] := 0.00;
BHML0 := 0.00;
BHML1 := 0.00;
EYHML[i] := 0.00;

end; {endfor}
for i := 1 to MAXNUM1 do begin
  RX[i] := 0.00;
end;
for i := 1 to MAXNUM1 do begin
(*  EX[i] := 0.00;*)
  EX2[i] := 0.00;
  EXY[i] := 0.00;
(*  ES[i] := 0.00;*)

```

```

ER[i] := 0.00;
EY[i] := 0.00;
EY2[i] := 0.00;

EH[i] := 0.00;
EYH[i] := 0.00;
ASE[i] := 0.00;
SE1[i] := 0.00;
ASE12[i] := 0.00;
SIJ2[i] := 0.00;
UIJ[i] := 0.00;
end; (endfor)

SUIJT := 0.00;
UT := 0.00;
EUI := 0.00;
SEUIJ := 0.00;
JACK1 := 0.00;

SL2 := 0.00;
LROLS := 0.00;
SUBL := 0.00;
SUBB := 0.00;
BART := 0.00;

(* RM[1] := 200; RV[1] := 1;
RM[2] := 200; RV[2] := 5;
RM[3] := 200; RV[3] := 50;
RM[4] := 200; RV[4] := 125;*)

Found_0 := False;
end; { }
procedure GenNumber;
var
  i : integer;

```

```

j   : integer;
k   : integer;
n   : integer;
ch  : char;
ZONE : real;
ZTWO : real;

begin { GENNUMBER }
(* while Seed = 0 do begin*)
(*   Seed := random(100);*)
(*   writeln('Seed ',Seed);*)
(* end; {endwhile} *)
for i := 1 to MAXNUM1 do begin
  RX[i] := rand(Seed);
end; {endfor}
j := 1;
for i := 1 to MAXNUM2 do begin
  ZONE := sqrt((-2)*ln(RX[j])) * cos(2*PI*RX[j+1]);
  ZTWO := sqrt((-2)*ln(RX[j])) * sin(2*PI*RX[j+1]);
(*   writeln('j ',j,' RX ',RX[j]);
  writeln('j+1 ',j+1,' RX ',RX[j+1]);
  writeln(' ', ' ZONE ',ZONE);
  writeln(' ', ' ZTWO ',ZTWO);*)

  case i of
    1..15 :begin
(*       EX[j] := (ZONE * RV[1]) + RM[1];*)
        if RX[i]-PI >0 then begin
          ER[j] := (ZONE * (sqrt(ES[j])))+em[j]
        end
        else begin
          ER[j] := (ZONE * (sqrt(ES[j])*C1))+em[j]
        end;
        EX2[j] := EX[j] * EX[j];
        EY[j] := B0 + (B1 * EX[j]) + ER[j];

```

```

EY2[j] := EY[j] * EY[j];
EXY[j] := EX[j] * EY[j];
SEX[1] := SEX[1] + EX[j];
SEX2[1] := SEX2[1] + EX2[j];
SEY[1] := SEY[1] + EY[j];
SEY2[1] := SEY2[1] + EY2[j];
SEXY[1] := SEXY[1] + EXY[j];
(*
EX[j+1] := (ZTWO * RV[1]) + RM[1]; *)
if RX[i]-P1 >0 then begin
    ER[j+1] := (Ztwo * (sqrt(ES[j+1]))) + em[j+1]
end
else begin
    ER[j+1] := (Ztwo * (sqrt(ES[j+1])*C1)) + em[j+1]
end;
EX2[j+1] := EX[j+1] * EX[j+1];
EY[j+1] := B0 + (B1 * EX[j+1]) + ER[j+1];
EY2[j+1] := EY[j+1] * EY[j+1];
EXY[j+1] := EX[j+1] * EY[j+1];
SEX[1] := SEX[1] + EX[j+1];
SEX2[1] := SEX2[1] + EX2[j+1];
SEY[1] := SEY[1] + EY[j+1];
SEY2[1] := SEY2[1] + EY2[j+1];
SEXY[1] := SEXY[1] + EXY[j+1];
j := j+2;
end;
16..30:begin
(*
EX[j] := (ZONE * RV[2]) + RM[2];*)
if RX[i]-P1 >0 then begin
    ER[j] := (ZONE * (sqrt(ES[j]))) + em[j]
end
else begin
    ER[j] := (ZONE * (sqrt(ES[j])*C1)) + em[j]
end;
EX2[j] := EX[j] * EX[j];

```

```

EY[j] := B0 + (B1 * EX[j]) + ER[j];
EY2[j] := EY[j] * EY[j];
EXY[j] := EX[j] * EY[j];
SEX[2] := SEX[2] + EX[j];
SEX2[2] := SEX2[2] + EX2[j];
SEY[2] := SEY[2] + EY[j];
SEY2[2] := SEY2[2] + EY2[j];
SEXY[2] := SEXY[2] + EXY[j];
(*
EX[J+1] := (ZTWO * RV[2]) + RM[2]; *)
if RX[i]-P1 >0 then begin
    ER[j+1] := (Ztwo * (sqrt(ES[j+1])))+EM[J+1]
end
else begin
    ER[j+1] := (Ztwo * (sqrt(ES[j+1])*C1))+EM[J+1]
end;
EX2[j+1]:= EX[j+1] * EX[j+1];
EY[j+1] := B0 + (B1 * EX[j+1]) + ER[j+1];
EY2[j+1]:= EY[j+1] * EY[j+1];
EXY[j+1]:= EX[j+1] * EY[j+1];
SEX[2] := SEX[2] + EX[j+1];
SEX2[2] := SEX2[2] + EX2[j+1];
SEY[2] := SEY[2] + EY[j+1];
SEY2[2] := SEY2[2] + EY2[j+1];
SEXY[2] := SEXY[2] + EXY[j+1];
j := j+2;
end;
31..45:begin
(*
EX[j] := (ZONE * RV[3]) + RM[3];*)
if RX[i]-P1 >0 then begin
    ER[j] := (ZONE * (sqrt(ES[j])))+EM[J]
end
else begin
    ER[j] := (ZONE * (sqrt(ES[j])*C1))+EM[J+1]
end;
end;

```

```

EX2[j] := EX[j] * EX[j];
EY[j] := B0 + (B1 * EX[j]) + ER[j];
EY2[j] := EY[j] * EY[j];
EXY[j] := EX[j] * EY[j];
SEX[3] := SEX[3] + EX[j];
SEX2[3] := SEX2[3] + EX2[j];
SEY[3] := SEY[3] + EY[j];
SEY2[3] := SEY2[3] + EY2[j];
SEXY[3] := SEXY[3] + EXY[j];
(*
EX[J+1] := (ZTWO * RV[3]) + RM[3]; *)
if RX[i]-P1 >0 then begin
    ER[j+1] := (Ztwo * (sqrt(ES[j+1]))) + EM[J+1]
end
else begin
    ER[j+1] := (Ztwo * (sqrt(ES[j+1])*C1)) + EM[J+1]
end;
EX2[j+1] := EX[j+1] * EX[j+1];
EY[j+1] := B0 + (B1 * EX[j+1]) + ER[j+1];
EY2[j+1] := EY[j+1] * EY[j+1];
EXY[j+1] := EX[j+1] * EY[j+1];
SEX[3] := SEX[3] + EX[j+1];
SEX2[3] := SEX2[3] + EX2[j+1];
SEY[3] := SEY[3] + EY[j+1];
SEY2[3] := SEY2[3] + EY2[j+1];
SEXY[3] := SEXY[3] + EXY[j+1];
j := j+2;
end;
46.60:begin
(*
EX[j] := (ZONE * RV[4]) + RM[4];*)
if RX[i]-P1 >0 then begin
    ER[j] := (ZONE * (sqrt(ES[j]))) + EM[J]
end
else begin
    ER[j] := (ZONE * (sqrt(ES[j])*C1)) + EM[J]

```

```

    end;
    EX2[j] := EX[j] * EX[j];
    EY[j] := B0 + (B1 * EX[j]) + ER[j];
    EY2[j] := EY[j] * EY[j];
    EXY[j] := EX[j] * EY[j];
    SEX[4] := SEX[4] + EX[j];
    SEX2[4] := SEX2[4] + EX2[j];
    SEY[4] := SEY[4] + EY[j];
    SEY2[4] := SEY2[4] + EY2[j];
    SEXY[4] := SEXY[4] + EXY[j];
    (*
    EX[j+1] := (ZTWO * RV[4]) + RM[4]; *)
    if RX[i]-P1 >0 then begin
        ER[j+1] := (Ztwo * (sqrt(ES[j+1]))) + EM[J+1]
    end
    else begin
        ER[j+1] := (Ztwo * (sqrt(ES[j+1])*C1)) + EM[J+1]
    end;
    EX2[j+1] := EX[j+1] * EX[j+1];
    EY[j+1] := B0 + (B1 * EX[j+1]) + ER[j+1];
    EY2[j+1] := EY[j+1] * EY[j+1];
    EXY[j+1] := EX[j+1] * EY[j+1];
    SEX[4] := SEX[4] + EX[j+1];
    SEX2[4] := SEX2[4] + EX2[j+1];
    SEY[4] := SEY[4] + EY[j+1];
    SEY2[4] := SEY2[4] + EY2[j+1];
    SEXY[4] := SEXY[4] + EXY[j+1];
    j := j+2;
end;
end; {endcase}
end; {endfor}
(* for i := 1 to MAXNUM1 do begin
    writeln('EX[';i:2;'] ',EX[i]);
    writeln('ES[';i:2;'] ',ES[i]);
    writeln('ER[';i:2;'] ',ER[i]);

```

```

        writeln('EX2[' ,i:2,'] ',EX2[i]);
        writeln('EY[' ,i:2,'] ',EY[i]);
        writeln('EY2[' ,i:2,'] ',EY2[i]);
        writeln('EXY[' ,i:2,'] ',EXY[i]);
    end; {endfor}*}
end; { GENNUMBER }
procedure GenEH;
var
    i : integer;
begin { GENEH }
    (* for i :=1 to 4 do begin
        writeln('SEX[' ,i,'] ',SEX[i]);
        writeln('SEX2[' ,i,'] ',SEX2[i]);
        writeln('SEY[' ,i,'] ',SEY[i]);
        writeln('SEY2[' ,i,'] ',SEY2[i]);
        writeln('SEXY[' ,i,'] ',SEXY[i]);
    end; {endfor}*}

    SXY := SEXY[1]+SEXY[2]+SEXY[3]+SEXY[4];
    SX  := SEX[1]+SEX[2]+SEX[3]+SEX[4];
    SY  := SEY[1]+SEY[2]+SEY[3]+SEY[4];
    SX2 := SEX2[1]+SEX2[2]+SEX2[3]+SEX2[4];

    BHAT1 := (SXY-((SX*SY)/MAXNUM1))/(SX2-((SX*SX)/MAXNUM1));
    BHAT0 := (SY/MAXNUM1) - (BHAT1*(SX/MAXNUM1));

    (* for i :=1 to 4 do begin
        writeln('SEX[' ,i,'] ',SEX[i]);
        writeln('SEX2[' ,i,'] ',SEX2[i]);
        writeln('SEY[' ,i,'] ',SEY[i]);
        writeln('SEY2[' ,i,'] ',SEY2[i]);
        writeln('SEXY[' ,i,'] ',SEXY[i]);
        writeln('BH0[' ,i,'] ',BH0[i]);
        writeln('BH1[' ,i,'] ',BH1[i]);
    end; {endfor}*}

```



```

end; (endfor)
*)
(* writeln('SXY ',SXY);
writeln('SX ',SX);
writeln('SY ',SY);
writeln('SX2 ',SX2);
writeln('BHAT0 ',BHAT0);
writeln('BHAT1 ',BHAT1);*)

for i := 1 to MAXNUM1 do begin
  case i of
    1..30 : begin
      EYH[i] := BHAT0 + (BHAT1*EX[i]);
      EH[i] := EY[i] - EYH[i];
      SSE[1] := SSE[1] + EH[i];
      SSEH[1]:= SSEH[1] + (EH[i] * EH[i]);
    end;
    31..60 : begin
      EYH[i] := BHAT0 + (BHAT1*EX[i]);
      EH[i] := EY[i] - EYH[i];
      SSE[2] := SSE[2] + EH[i];
      SSEH[2]:= SSEH[2] + (EH[i] * EH[i]);
    end;
    61..90 : begin
      EYH[i] := BHAT0 + (BHAT1*EX[i]);
      EH[i] := EY[i] - EYH[i];
      SSE[3] := SSE[3] + EH[i];
      SSEH[3]:= SSEH[3] + (EH[i] * EH[i]);
    end;
    91..120 : begin
      EYH[i] := BHAT0 + (BHAT1*EX[i]);
      EH[i] := EY[i] - EYH[i];
      SSE[4] := SSE[4] + EH[i];
      SSEH[4]:= SSEH[4] + (EH[i] * EH[i]);
    end;
  end;
end;

```

```

    end; {endcase}
end; {endfor}

/* for i := 1 to MAXNUM1 do begin
    writeln('EY[',i,'] ',EY[i]);
    writeln('EYH[',i,'] ',EYH[i]);
    writeln('EH[',i,'] ',EH[i]);
end; {endfor}*)

for i := 1 to 4 do begin
    SSEH[i] := (SSEH[i] / IX);
(*    writeln('SSEH[i] ',SSEH[i]);*)
end; {endfor}

temp1 := 0.00;
temp2 := 0.00;
temp3 := 0.00;
temp4 := 0.00;
temp5 := 0.00;
temp6 := 0.00;
temp7 := 0.00;
temp8 := 0.00;
temp9 := 0.00;

(* temp1 := ( IX * ( (1/SSEH[1])+(1/SSEH[2])+(1/SSEH[3])+(1/SSEH[4]) ) );
temp2 := ( (SEX[1]/SSEH[1])+(SEX[2]/SSEH[2])+(SEX[3]/SSEH[3])+(SEX[4]/SSEH[4]) );
temp3 := ( (SEX2[1]/SSEH[1])+(SEX2[2]/SSEH[2])+(SEX2[3]/SSEH[3])+(SEX2[4]/SSEH[4]) );
temp4 := ( 1/( (temp1*temp3)-(temp2*temp2) ) );
temp5 := ( temp3*temp4 );
temp6 := ( ((-1)*(temp2))*temp4 );
temp7 := ( temp1*temp4 );
temp8 := ( (SEY[1]/SSEH[1])+(SEY[2]/SSEH[2])+(SEY[3]/SSEH[3])+(SEY[4]/SSEH[4]) );
temp9 := ( (SEXY[1]/SSEH[1])+(SEXY[2]/SSEH[2])+(SEXY[3]/SSEH[3])+(SEXY[4]/SSEH[4]) );
BHML0 := ( (temp5*temp8) + (temp6*temp9) );
BHML1 := ( (temp6*temp8) + (temp7*temp9) );*)

(* writeln('BHML0 ',BHML0);

```

```

writeln('BHML1 ',BHML1;*)
(* for i := 1 to maxnum1 do begin
  case i of
    1..30 : begin
      EYHML[i] := BHML0 + (BHML1*EX[i]);
      EHML[i] := EY[i] - EYHML[i];
      SSEHT[1] := SSEHT[1] + (EHML[i] * EHML[i]);
    end;
    31..60 : begin
      EYHML[i] := BHML0 + (BHML1*EX[i]);
      EHML[i] := EY[i] - EYHML[i];
      SSEHT[2] := SSEHT[2] + (EHML[i] * EHML[i]);
    end;
    61..90 : begin
      EYHML[i] := BHML0 + (BHML1*EX[i]);
      EHML[i] := EY[i] - EYHML[i];
      SSEHT[3] := SSEHT[3] + (EHML[i] * EHML[i]);
    end;
    91..120 : begin
      EYHML[i] := BHML0 + (BHML1*EX[i]);
      EHML[i] := EY[i] - EYHML[i];
      SSEHT[4] := SSEHT[4] + (EHML[i] * EHML[i]);
    end;
  end; {endcase}
end; {endfor}*)
(* for i := 1 to 4 do begin
  SSEHT[i] := (SSEHT[i] / IX);
  writeln('SSEHT[i] ',SSEHT[i]);
end; {endfor}*)
end; { GENEH }
procedure GenSE;
var i : integer;
begin { GENSE }
  for i :=1 to 4 do begin

```

```

(*   writeln('SSE[i] ',SSE[i]);*)
    SE[i] := SSE[i]/IX;
(*   writeln(' SE[i] ',SE[i]);   *)
end; {endfor}
end; { GENSE }
procedure CalSE;
var i : integer;
    temp : real;
begin { CALSE }
  for i := 1 to MAXNUM1 do begin
    case i of
      1..30 :begin
        temp := 0.00;
        temp := EH[i] - SE[1];
        SSI[1] := SSI[1] + (temp * temp);
        end;
      31..60 :begin
        temp := 0.00;
        temp := EH[i] - SE[2];
        SSI[2] := SSI[2] + (temp * temp);
        end;
      61..90 :begin
        temp := 0.00;
        temp := EH[i] - SE[3];
        SSI[3] := SSI[3] + (temp * temp);
        end;
      91..120 :begin
        temp := 0.00;
        temp := EH[i] - SE[4];
        SSI[4] := SSI[4] + (temp * temp);
        end;
    end; {endcase}
  end; {endfor}
  for i := 1 to 4 do begin

```

```

(*   writeln('SSI[',i,'] ',SSI[i]);*)
  SI2[i] := SSI[i]/VH1;
(*   writeln('SI2[',i,'] ',SI2[i]);   *)
end; {endfor}
end; { CASE }
procedure AdjustSE;
  var i :integer;
      j :integer;
      temp : real;
begin { ADJUSTSE }
  (or i := 1 to MAXNUM) do begin
    case i of
      1..30:begin
        for j := 1 to 30 do begin
          if i <> j then begin
            ASE[i] := ASE[i] + EH[j];
          end else begin
            end; {endif}
          end; {endfor}
        end;
      31..60:begin
        for j := 31 to 60 do begin
          if i <> j then begin
            ASE[i] := ASE[i] + EH[j];
          end else begin
            end; {endif}
          end; {endfor}
        end;
      61..90:begin
        for j := 61 to 90 do begin
          if i <> j then begin
            ASE[i] := ASE[i] + EH[j];
          end else begin
            end; {endif}
          end; {endfor}
        end;
      end;
    end;
  end;
end;

```

```

        end; (endfor)
    end;
    91..120:begin
        for j := 91 to 120 do begin
            if i <> j then begin
                ASE[i] := ASE[i] + EH[j];
            end else begin
                end; (endif)
            end; (endfor)
        end;
    end; (endcase)
end; (endfor)
for i := 1 to MAXNUM1 do begin
    SE1[i] := ASE[i]/(IX-1);
    (*   writeln('SE1[';i;'] ',SE1[i]);*)
end; (endfor)
for i := 1 to MAXNUM1 do begin
    case i of
        1..30 : begin
            for j := 1 to 30 do begin
                if i <> j then begin
                    temp := 0.00;
                    temp := EH[j] - SE1[i];
                    ASE12[i] := ASE12[i] + (temp * temp);
                end else begin
                    end; (endif)
                end; (endfor)
            end;
        31..60 : begin
            for j := 31 to 60 do begin
                if i <> j then begin
                    temp := 0.00;
                    temp := EH[j] - SE1[i];
                    ASE12[i] := ASE12[i] + (temp * temp);
                end else begin
                    end; (endif)
                end; (endfor)
            end;
        end;
    end;
end;

```

```

        end else begin
        end; {endif}
    end; {endfor}
end;

61..90 : begin
    for j := 61 to 90 do begin
        if i <> j then begin
            temp := 0.00;
            temp := EH[j] - SE1[i];
            ASE12[i] := ASE12[i] + (temp * temp);
        end else begin
        end; {endif}
    end; {endfor}
end;

91..120 : begin
    for j := 91 to 120 do begin
        if i <> j then begin
            temp := 0.00;
            temp := EH[j] - SE1[i];
            ASE12[i] := ASE12[i] + (temp * temp);
        end else begin
        end; {endif}
    end; {endfor}
end;

end; {endcase}
end; {endfor}

for i := 1 to MAXNUM1 do begin
    SIJ2[i] := ASE12[i] / VI2;
    (* writeln('SIJ2['.i.'].SIJ2[i]); *)
end; {endfor}

end; { ADJUSTSE }

procedure CalSL2;

var
    i : integer;

```

```

begin { CAL_SL2 }
  SL2 := 0.00;
  for i :=1 to MAXNUM1 do begin
    SL2 := SL2 +(EH[i]*EH[i]);
    SLML2 := SLML2 + (EH[i]*EH[i]);
  end; {endfor}
  SL2 := SL2 /(MAXNUM1 -2);
  SLML2 := SLML2/(MAXNUM1);
  (* writeln('SL2 ',SL2);
  writeln('SLML2 ',SLML2);*)
end; { CAL_SL2 }

procedure Check_S2;
var
  i : integer;
begin { CHECK_S2 }
  Found_0 := False;
  for i := 1 to MAXNUM1 do begin
    if SIJ2[i] = 0.00 then begin
      Found_0 := True;
      i := MAXNUM1 +1;
    end else begin

    end; {endif}
  end; {endfor}
  for i := 1 to 4 do begin
    if SI2[i] = 0.00 then begin
      Found_0 := True;
      i := 5;
    end else begin
    end; {endif}
  end; {endfor}
  if SL2 = 0.00 then begin
    Found_0 := True;
  end else begin

```



```

end; {endif}

end; { CHECK_S2 }

procedure CalJACK;
var
  i :integer;
  temp : real;
begin { CALJACK }
  for i := 1 to MAXNUM1 do begin
    case i of
      1..30 :begin
        UIJ[i] := (IX * ln(SI2[1])) -((IX-1) * ln(SIJ2[i]));
        SUIJ[1]:= SUIJ[1] + UIJ[i];
        SUIJT := SUIJT + UIJ[i];
      end;
      31..60 :begin
        UIJ[i] := (IX * ln(SI2[2])) -((IX-1) * ln(SIJ2[i]));
        SUIJ[2]:= SUIJ[2] + UIJ[i];
        SUIJT := SUIJT + UIJ[i];
      end;
      61..90 :begin
        UIJ[i] := (IX * ln(SI2[3])) -((IX-1) * ln(SIJ2[i]));
        SUIJ[3]:= SUIJ[3] + UIJ[i];
        SUIJT := SUIJT + UIJ[i];
      end;
      91..120 :begin
        UIJ[i] := (IX * ln(SI2[4])) -((IX-1) * ln(SIJ2[i]));
        SUIJ[4]:= SUIJ[4] + UIJ[i];
        SUIJT := SUIJT + UIJ[i];
      end;
    end; {endcase}
  end; {endfor}

  for i := 1 to 4 do begin

```

```

    UI[i] := SUIJ[i] / IX;
end; {endfor}
UT := SUIJT / MAXNUM1;
for i := 1 to 4 do begin
    temp := 0.00;
    temp := IX * ((UI[i] - UT) * (UI[i] - UT));
    EUI := EUI + temp;
end; {endfor}
for i := 1 to MAXNUM1 do begin
    case i of
        1..30 : begin
            temp := 0.00;
            temp := UIJ[i] - UI[1];
            EUJ[1] := EUJ[1] + (temp * temp);
        end;
        31..60 : begin
            temp := 0.00;
            temp := UIJ[i] - UI[2];
            EUJ[2] := EUJ[2] + (temp * temp);
        end;
        61..90 : begin
            temp := 0.00;
            temp := UIJ[i] - UI[3];
            EUJ[3] := EUJ[3] + (temp * temp);
        end;
        91..120 : begin
            temp := 0.00;
            temp := UIJ[i] - UI[4];
            EUJ[4] := EUJ[4] + (temp * temp);
        end;
    end; {endcase}
end; {endfor}
for i := 1 to 4 do begin
    SEUIJ := SEUIJ + EUJ[i];

```

```

end; {endfor}
JACK1 := (EUI/3) / (SEUI/(MAXNUM1-4));
J_1 := 0;
J_5 := 0;
if JACK1 > F_1_3_116 then begin
  Rej_J_1 := Rej_J_1 + 1;
  (* writeln('Reject JACK1 .01 > F ',JACK1);*)
  J_1 := 1;
end else begin
  Acc_J_1 := Acc_J_1 + 1;
  (* writeln('Accept JACK1 .01 < F ',JACK1); *)
  J_1 := 0;
end; {endif}
(* writeln('Reject Time ',Rej_J_1);
writeln('Accept Time ',Acc_J_1);*)
if JACK1 > F_5_3_116 then begin
  Rej_J_5 := Rej_J_5 + 1;
  (* writeln('Reject JACK1 .05 > F ',JACK1);*)
  J_5 := 1;
end else begin
  Acc_J_5 := Acc_J_5 + 1;
  (* writeln('Accept JACK1 .05 < F ',JACK1);*)
  J_5 := 0;
end; {endif}
(* writeln('Reject Time ',Rej_J_5);
writeln('Accept Time ',Acc_J_5);*)
end; { CALJACK }
procedure CalLike;
var i : integer;
begin { CALLIKE }
  for i := 1 to 4 do begin
    SUBL := SUBL + (IX *ln(SI2[i]));
  end; {endfor}
  LROLS := (MAXNUM1 * (ln(SL2))) - SUBL;

```

```

L_1 := 0;
L_5 := 0;
if LROLS > X_5_3 then begin
  Rej_L_5 :=Rej_L_5 + 1;
  (*  writeln('Reject LROLS .05 > X2 ',LROLS);  *)
  L_5 := 1;
end else begin
  Acc_L_5 :=Acc_L_5 + 1;
  (*  writeln('Accept LROLS .05 < X2 ',LROLS);*)
  L_5 := 0;
end; {endif}
(*  writeln('Reject Time ',Rej_L_5);
  writeln('Accept Time ',Acc_L_5);*)
if LROLS > X_1_3 then begin
  Rej_L_1 :=Rej_L_1 + 1;
  (*  writeln('Reject LROLS .01 > X2 ',LROLS);*)
  L_1 := 1;
end else begin
  Acc_L_1 :=Acc_L_1 + 1;
  (*  writeln('Accept LROLS .01 < X2 ',LROLS);*)
  L_1 := 0;
end; {endif}
(*  writeln('Reject Time ',Rej_L_1);
  writeln('Accept Time ',Acc_L_1);  *)
end; { CALLIKE }
procedure CalBart;
var
  i : integer;
  temp : real;
begin { CALBART }
  for i :=1 to 4 do begin
    SUBB := SUBB + ((IX-1) *ln(SI2[i]));
  end; {endfor}
  temp := 0.00;

```

```

temp := 1+( 1/9*( ((1/(IX-1)) + (1/(IX-1)) + (1/(IX-1)) + (1/(IX-1))-(1/(MAXNUM1-4))) ));
BART := ( (((MAXNUM1 -4)*(ln(SL2))) - SUBB) / temp );
B_1 := 0;
B_5 := 0;
if BART > X_5_3 then begin
    Rej_B_5 :=Rej_B_5 + 1;
    (*    writeln('Reject BART .05 > X2 ',BART);    *)
    B_5 := 1;
end else begin
    Acc_B_5 :=Acc_B_5 + 1;
    (*    writeln('Accept BART .05 < X2 ',BART);*)
    B_5 := 0;
end; (endif)
(*    writeln('Reject Time ',Rej_B_5);
    writeln('Accept Time ',Acc_B_5);    *)
if BART > X_1_3 then begin
    Rej_B_1 :=Rej_B_1 + 1;
    (*    writeln('Reject BART .01 > X2 ',BART);*)
    B_1 := 1;
end else begin
    Acc_B_1 :=Acc_B_1 + 1;
    (*    writeln('Accept BART .01 < X2 ',BART);*)
    B_1 := 0;
end; (endif)
(*    writeln('Reject Time ',Rej_B_1);
    writeln('Accept Time ',Acc_B_1);*)
end; ( CALBART )
begin { }

Cnt_Loop := 1; (* count run *)
for i := 1 to MAXNUM1 do begin
    EX[i] := 0.00;
    RX[i] := 0.00;
    ES[i] := 0.00;

```

```

    EM[I] := 0.00;
end; {endfor}

RM[1] := 200.0; RV[1] := 2.0;
RM[2] := 300.0; RV[2] := 3.0;
RM[3] := 400.0; RV[3] := 4.0;
RM[4] := 500.0; RV[4] := 5.0;

seed :=0;
randomize;
Randdf;
Seed := Random(100);

for i := 1 to MAXNUM1 do begin
    RX[i] := rand(Seed);
end; {endfor}

j := 1;
for i := 1 to MAXNUM2 do begin
    ZONE := sqrt((-2)*ln(RX[j])) * cos(2*PI*RX[j+1]);
    ZTWO := sqrt((-2)*ln(RX[j])) * sin(2*PI*RX[j+1]);
    case i of
        1..15 :begin
            EX[j] := (ZONE * RV[1]) + RM[1];
            EX[J+1] := (ZTWO * RV[1]) + RM[1];
            j := j+2;
        end;
        16..30:begin
            EX[j] := (ZONE * RV[2]) + RM[2];
            EX[J+1] := (ZTWO * RV[2]) + RM[2];
            j := j+2;
        end;
        31..45:begin
            EX[j] := (ZONE * RV[3]) + RM[3];
            EX[J+1] := (ZTWO * RV[3]) + RM[3];
            j := j+2;
        end;
    end;
end;

```

```

        end;
    46..60:begin
        EX[j] := (ZONE * RV[4]) + RM[4];
        EX[j+1] := (ZTWO * RV[4]) + RM[4];
        j := j+2;
    end;
end; {endcase}
end; {endfor}
for i:= 1 to MAXNUM1 do begin
    case i of
        1..30:begin
            ES[i] :=1;
            EM[I] :=100;
        end;
        31..60:begin
            ES[i] :=5;
            EM[I] :=100;
        end;
        61..90:begin
            ES[i] :=10;
            EM[I] :=100;
        end;
        91..120:begin
            ES[i] :=15;
            EM[I] :=100;
        end;
    end; {endcase}
end; {endfor}

Acc_J_1 := 0;  Rej_J_1 := 0;
Acc_L_1 := 0;  Rej_L_1 := 0;
Acc_B_1 := 0;  Rej_B_1 := 0;
Acc_J_5 := 0;  Rej_J_5 := 0;
Acc_L_5 := 0;  Rej_L_5 := 0;

```

```

Acc_B_5 := 0;  Rej_B_5 := 0;
Seed :=0;
Cnt_exp :=1; (* total count run *)
randomize;
Randdf;
while Cnt_exp <=5 do begin
  seed :=0;
  Cnt_loop :=1;
  while Seed = 0 do begin
    Seed := Random(100);
  end; {endwhile}
while Cnt_Loop <= 100 do begin
  InitialVar;
  GenNumber;
  GenEH;
  GenSE;
  CalSE;
  AdjustSE;
  CalSL2;
  Check_S2;
  if Found_0 then begin
  end else begin
    CalJACK;
    CalLike;
    CalBart;
    Cnt_Seq1 := ((Cnt_Loop -1) * MAXNUM1);
    Cnt_Seq2 := ((Cnt_Loop -1) * 4    );
    Cnt_Seq3 := (Cnt_Loop -1);
    inc(Cnt_Loop);
  end; {endif}
end; {endwhile 2}
inc(Cnt_exp);
end; {endwhile 1}
writeln('Reject Time JACK 0.01 '.Rej_J_1);

```



```
writeln('Reject Time LROLS 0.01 ',Rej_L_1);  
writeln('Reject Time BART 0.01 ',Rej_B_1);  
writeln;  
writeln('Reject Time JACK 0.05 ',Rej_J_5);  
writeln('Reject Time LROLS 0.05 ',Rej_L_5);  
writeln('Reject Time BART 0.05 ',Rej_B_5);  
end. { }
```

ประวัติผู้เขียน

นางสาวรุ่งอรุณ เตชะทรงชัย เกิดเมื่อวันที่ 5 มีนาคม 2511 สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต (วท.บ.) สาขาสถิติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2533 และเข้าศึกษาต่อในหลักสูตร วิทยาศาสตรมหาบัณฑิต ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2534

