

บรรณานุกรม

ภาษาไทย

หนังสือ

รสสุคนธ์ หังสพฤกษ์. การเขียนโปรแกรมภาษาปาสคาล. กรุงเทพมหานคร :

สำนักพิมพ์ไต้เต็คบุ๊กส์ , 2528.

เอกสารอื่น ๆ

สมชัย ยืนนาน. "การศึกษาโดยวิธีมอนติคาร์โลเปรียบเทียบอำนาจของการทดสอบของการเท่ากันของความแปรปรวนระหว่างประชากรสองกลุ่ม" วิทยานิพนธ์ปริณายมหาบัณฑิต แผนกวิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2528.

สมพิศ โชติวิทยากร. "การเปรียบเทียบอำนาจการทดสอบของตัวสถิติบางตัวที่ใช้ในการทดสอบการแจกแจงปกติ" วิทยานิพนธ์ปริณายมหาบัณฑิต แผนกวิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย , 2531.

ภาษาต่างประเทศ

หนังสือ

Gibbons, J.D. Nonparametric Statistical Inference. Tokyo: Mc Graw Hill, 1971.

Pearson, E.S. and Hartley, H.O. Biometrika Tables for Statisticians, Vol.II. London : Cambridge University Press , 1972.

Peter, James F. Pascal With Program Design. New York : Holt Rinehart and Winston; 1986.

Morgan, Byron J.T. Elements of Simulation. London Chapman and Hall, 1984.

บทความ

- Ajne, B. " A simple test for uniformity of a circular distribution."
Biometrika 55 (January 1968): 343-354.
- Darling, D.A. " On a class of problem related to the random devision
of an interval." Ann. Math. Stat.24 (1953): 239-253.
- Durbin, J. " Some method of an constructing exact tests. Biometrika 48
(April 1961) : 41-55.
- Pearson, E.S. " Comparison of tests for randomness of points on a line ."
Biometrika 50.(1963) : 315-325.
- Pearson, E.S. and Stephens, M.A. " The Goodness-of-fit tests based on
 W^2_N and U^2_N ." Biometrika 49 (1962) : 397-402.
- Ramsey, P.H. " Exact Type I Error Rates for Robustness of Student's t
Tests with unequal Variances." Journal of Education Statistics 5
(Winter 1980) : 337 -349..
- Stephens, M.A. " The distribution of the goodness-of-fit statistic U_N^2 I."
Biometrika 50 (1963) : 303 - 313.
- Stephens, M.A. " The distribution of the goodnees-of-fit statistic, U_N^2 II."
Biometrika 51 (1964) : 393-397.
- Stephens, M.A. " Use of the Kolmogorov-Smirov, Cramer-Von Mises and Related
Statistics without Extensive Tables." Journal of the Royal
Statistical Society , Ser.B, 32 ,
No.1 (1970) : 115-122.
- Stephens, M.A. " .EDF Statistics for goodness of Fit and Some Comparisons."
Journal of the American Statistical Association 69
(September 1974) : 730-737.
- Watson, G.S. " Goodness-of-fit tests on a circle.I." Biometrika 48
(1961) : 109-114.

Watson, G.S. " Goodness-of-fit tests on a circle .II."

Biometrika 49 (1962) : 57-63.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การแจกแจงแบบยูนิฟอร์มในช่วง (0,1)

ถ้า X เป็นตัวแปรสุ่มที่มีการแจกแจงแบบยูนิฟอร์มในช่วง $(0,1)$ จะได้ค่าเฉลี่ย ($E(x)$) และค่าแปรปรวนของ X ($V(x)$) ดังนี้.

$$E(x) = 1/2$$

$$V(x) = 1/12$$

การตรวจสอบว่าข้อมูลที่ต้องการศึกษามีการแจกแจงแบบยูนิฟอร์มในช่วง (0,1)

ในที่นี้จะทำการตรวจสอบข้อมูลทั้ง 4 ลักษณะที่สร้างขึ้นโดยใช้โปรแกรมภาษาปาสคาล โดยใช้ procedure ต่างๆ ดังนี้

1. procedure generate data
2. procedure transform 1
3. procedure transform 2
4. procedure transform 3

โดยจะทำการตรวจสอบว่าข้อมูลที่ได้จาก procedure ทั้ง 4 แบบดังกล่าวนี้จะมีการแจกแจงแบบยูนิฟอร์มในช่วง $(0,1)$ หรือไม่ โดยที่สำหรับข้อมูลที่ได้จากการแปลงทั้ง 3 รูปแบบนั้นจะทำการทดสอบทุกค่าของ $j = 1,2,3,4$ ในทุกรูปแบบของการแปลง

สำหรับการตรวจสอบในที่นี้จะใช้ Pearson X^2 Statistic โดยกำหนดกลุ่มของข้อมูลเป็น 8 กลุ่ม นั่นคือทำการทดสอบโดยใช้ตัวสถิติ X^2 นั้นเอง และกำหนดให้ขนาดตัวอย่างเท่ากับ 50 และใช้ระดับนัยสำคัญในการทดสอบเป็น 0.01 และ 0.05 และการทดสอบนี้จะใช้ procedure test uniform

สมมติฐานในการทดสอบคือ

H_0 : ประชากรมีการแจกแจงแบบยูนิฟอร์มในช่วง (0,1)

H_a : ประชากรไม่มีการแจกแจงแบบยูนิฟอร์มในช่วง (0,1)

ผลการทดสอบปรากฏดังตาราง

Y	J	X8
0	0	3.8333
1	1	6.3333
	2	9.0000
	3	11.3333
	4	14.1667
2	1	5.0000
	2	6.3333
	3	10.3333
	4	12.6667
3	1	4.0000
	2	3.6667
	3	5.1667
	4	9.1667

ขอบเขตวิกฤต เมื่อระดับนัยสำคัญ เป็น 0.01 จะปฏิเสธสมมติฐาน H_0 เมื่อ

ค่า X8 ที่คำนวณได้ > 18.50 และ

เมื่อระดับนัยสำคัญ เป็น 0.05 จะปฏิเสธสมมติฐาน H_0 เมื่อ

ค่า X8 ที่คำนวณได้ > 14.10

จากผลการทดสอบจะเห็นว่าถ้ากำหนดระดับนัยสำคัญเป็น 0.01 ก็จะสรุปได้ว่าผลการทดสอบจะยอมรับสมมติฐาน H_0 สำหรับทุกรูปแบบของข้อมูล แต่ถ้ากำหนดระดับนัยสำคัญเป็น 0.05 ก็จะสรุปได้ว่าผลการทดสอบจะยอมรับสมมติฐาน H_0 เกือบทุกรูปแบบของข้อมูล ยกเว้น เฉพาะเมื่อ $Y=1, j=4$ ซึ่งค่า X8 ที่คำนวณได้มีค่าเท่ากับ 14.1667 ซึ่งมากกว่า 14.10

เพียงเล็กน้อยเท่านั้นดังนั้นจึงสามารถสรุปได้ว่าข้อมูลที่ได้จากการ generate และ ข้อมูลที่ได้จากการ transform ทั้ง 3 รูปแบบ สำหรับทุกค่าของ $j = 1, 2, 3, 4$ มีการแจกแจงแบบยูนิฟอร์มในช่วง $(0, 1)$

จากนั้นก็ทำการคำนวณค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และค่าสัมประสิทธิ์ความแปรผันของข้อมูลทุกรูปแบบที่ต้องการศึกษา เพื่อพิจารณาว่าข้อมูลที่ได้จากการ transform ทั้ง 3 รูปแบบ สำหรับทุกค่าของ $j = 1, 2, 3, 4$ นั้นมีค่าสัมประสิทธิ์ความแปรผันเป็นเท่าไรบ้าง ซึ่งการคำนวณในขั้นนี้จะใช้ procedure mean and SD ผลการคำนวณเป็นดังนี้

```

Filename : N50Y0J0.DAT
=====> Mean and SD =
          Mean of population is    0.498518
          SD of population is      0.288207
          CV of population is      0.578127

```

```

Filename : N50Y1J1.DAT
=====> Mean and SD =
          Mean of population is    0.331581
          SD of population is      0.297054
          CV of population is      0.895870

```

```

Filename : N50Y1J2.DAT
=====> Mean and SD =
          Mean of population is    0.248160
          SD of population is      0.282204
          CV of population is      1.137186

```

```

Filename : N50Y1J3.DAT
=====> Mean and SD =
          Mean of population is    0.198184
          SD of population is      0.265302
          CV of population is      1.338668

```

```

Filename : N50Y1J4.DAT
=====> Mean and SD =
          Mean of population is    0.164931
          SD of population is      0.249798
          CV of population is      1.514564

```

Filename : N50Y2J1.DAT

=====> Mean and SD =

Mean of population is 0.498711
SD of population is 0.208498
CV of population is 0.418074

Filename : N50Y2J2.DAT

=====> Mean and SD =

Mean of population is 0.494945
SD of population is 0.175935
CV of population is 0.355464

Filename : N50Y2J3.DAT

=====> Mean and SD =

Mean of population is 0.498955
SD of population is 0.149801
CV of population is 0.300229

Filename : N50Y2J4.DAT

=====> Mean and SD =

Mean of population is 0.500489
SD of population is 0.138448
CV of population is 0.276626

Filename : N50Y3J1.DAT

=====> Mean and SD =

Mean of population is 0.501160
SD of population is 0.341475
CV of population is 0.681370

Filename : N50Y3J2.DAT

=====> Mean and SD =

Mean of population is 0.496450
SD of population is 0.378222
CV of population is 0.761854

Filename : N50Y3J3.DAT

=====> Mean and SD =

Mean of population is 0.497604
SD of population is 0.401840
CV of population is 0.807551

Filename : N50Y3J4.DAT

=====> Mean and SD =

Mean of population is 0.500717
SD of population is 0.417940
CV of population is 0.834684

หมายเหตุ ผลการคำนวณได้สรุปไว้แล้วดังตารางที่ 3.1



ภาคผนวก ข

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

(*

PROGRAM TO COMPUTE
PROBABILITY OF TYPE 1 ERROR AND POWER OF THE TEST

101

DESCRIPTION OF VARIABLE

N,N1 = Number of Observe
MEAN = Mean of Observe Each Case
RMEAN = Mean of Observe
SD = Standard Deviation of Observe
CV = Coefficient of variation

DESCRIPTION OF FUNCTION AND PROCEDURE

function power(a : real;b : integer):real;
o Use for Find the Result of A with Power B (A Power B)
procedure input_datafile;
o Use for Open Data File Which be Process
procedure output_datafile;
o Use for Open Output File Which The Result of Process
will be write
procedure transform_data;
o Use for Main Procedure to Call Transform Data
procedure transform1;
o Use for Transform Data Which Y=1, It was use with J In
Transform_data Procedure
procedure transform2;
o Use for Transform Data Which Y=2, It was use with J In
Transform_data Procedure
procedure transform3;
o Use for Transform Data Which Y=3, It was use with J In
Transform_data Procedure
procedure generate_data;
o Use for Generate Data Which be Process
procedure printreport;
o Use for Write the report for Transfer to LOTUS-123
procedure chi_square(n : integer;var f1,f2 : boolean);
o Use for Compute Peason Chi Square Test 8 Level
procedure chi_square1(n : integer;var f1,f2 : boolean);
o Use for Compute Peason Chi Square Test 5 Level
procedure test_d(n : integer;var f1,f2 : boolean);
o Use for Compute Kolmogorov Smirnov Test
procedure test_U_square(n : integer;var f1,f2 : boolean);
o Use for Compute Watson Test
procedure test_A(n : integer;var f1,f2 : boolean);
o Use for Compute Anderson Darling Test
procedure test_Q(n : integer;var f1,f2 : boolean);
o Use for Compute Q Statistic Test
procedure test_uniformity
o Use for test that data is U(0,1) or not
procedure mean_and_SD(n: integer);
o Use for Compute Mean, SD and CV

*)

```

program Test_statistics_with_pull_down_menu;
(*                                                                    *)
(*          Use the Standard Procedure                               *)
(*                                                                    *)
uses crt,screen,Keyboard,win,ton;

const
  Maxchoice = 7;  (* Maximum Choice of Pull Down Menu *)
  MaxMenu = 2;   (* Maximum Menu of Pull Down Menu *)

type

  r = record          (* Record Layout of Data File *)
    data : array[1..100] of real;
  end;
(*                                                                    *)
(*          Record Layout of Report Which will Transfer to LOTUS-123 *)
(*                                                                    *)
  r1 =record
    p : array [1..2,1..7,0..3,0..4,1..9] of real;
  end;
  string6 = string[6];
  string80 = string[80];
  structuremenu = record
    win : array[1..4] of byte;
    col : array[0..Maxchoice] of byte;
    Row : array[0..Maxchoice] of byte;
    msg : array[0..Maxchoice] of string80;
    lastchoice : byte;
  end;

  strucmenu = array[1..Maxmenu] of Structuremenu;
  FunctionMenu = Record
    Col : Array[1..10] of byte;
    Row : Array[1..10] of byte;
    Msg : Array[1..10] of string6;
  end;

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย


```

(*)
(*)          Display Function Key Available          (*)
(*)
procedure displayfunkey;
var   i : byte;
begin
  window(1,1,80,25);
  with Fkey do begin
    for i:=1 to 10 do begin
      gotoxy(col[i],Row[i]);
      setattr(lowdisplay);
      write(i);
      SetAttr(reverselow);
      write(msg[i]);
    end;
  end;
end;

(*)
2*          Display Main Menu                      (*)
(*)
procedure displayMainmenu;
var   i : byte;
begin
  window(1,1,80,25);
  setattr(lowdisplay);
  for i:=1 to maxmenu do
    with menu[i] do begin
      gotoxy(col[0],row[0]);
      write(msg[0]);
    end;
  end;
end;

(*)
(*)          Change the Hightlight to New Active Choice (*)
(*)
procedure choiceactive(old,new: byte);
begin
  with menu[statusmenu] do begin
    setattr(lowdisplay);
    gotoxy(col[old],row[old]);
    write(msg[old]);
    setattr(reverselow);
    gotoxy(col[new],row[new]);
    write(msg[new]);
  end;
end;

```

```

(*)
(*) Change Active Menu (*)
(*)
procedure menuactive(new: byte);
var
    i : byte;
begin
    displaymainmenu;
    setattr(reversehigh);
    with menu[new] do begin
        gotoxy(col[0],row[0]);
        write(msg[0]);
        setwinattr(lowdisplay);
        setboxattr(highdisplay);
        setcharattr(lowdisplay);
        setboxstyle(double);
        setwinheader('');
        windowopen(win[1],win[2],win[3],win[4]);
        for i:=1 to lastchoice do begin
            gotoxy(col[i],row[i]);
            write(msg[i]);
        end;
    end;
    choiceactive(1,choicemenu[new]);
end;

(*)
(*) Move the HightLight Up (*)
(*)
procedure moveup;
var
    currentchoice : byte;
begin
    currentchoice:=choicemenu[statusmenu];
    if currentchoice =1 then
        choicemenu[statusmenu]:=menu[statusmenu].lastchoice
    else
        choicemenu[statusmenu]:=currentchoice-1;
        choiceactive(currentchoice,choicemenu[statusmenu]);
end;

(*)
(*) Move the HightLight Down (*)
(*)
procedure movedown;
var
    currentchoice : byte;
begin
    currentchoice:=choicemenu[statusmenu];
    if currentchoice = menu[statusmenu].lastchoice then
        choicemenu[statusmenu]:=1
    else
        choicemenu[statusmenu]:=currentchoice+1;
        choiceactive(currentchoice,choicemenu[statusmenu]);
end;

```

```

(*)
(*)           Move the Active Menu to the Next Menu      (*)
(*)
procedure moveforward;
begin
  windowclose;
  if statusmenu+1>maxmenu then statusmenu:=1
    else statusmenu:=statusmenu+1;
  menuactive(statusmenu);
end;

(*)
(*)           Move the Active Menu to Last Menu          (*)
(*)
procedure moveback;
begin
  windowclose;
  if statusmenu=1 then statusmenu:=maxmenu
    else statusmenu:=statusmenu-1;
  menuactive(statusmenu);
end;

(*)
(*)           Move the HightLight to the First Choice of Menu (*)
(*)
procedure movetofirstchoice;
begin
  choiceactive(choicemenu[statusmenu],1);
  choicemenu[statusmenu]:=1;
end;

(*)
(*)           Move the HightLight to the Last Choice of Menu (*)
(*)
procedure movetolastchoice;
begin
  with menu[statusmenu] do begin
    choiceactive(choicemenu[statusmenu],lastchoice);
    choicemenu[statusmenu]:=lastchoice;
  end;
end;

(*)
(*)           Move Active Menu to First Menu             (*)
(*)
procedure movetofirstmenu;
begin
  windowclose;
  statusmenu:=1;
  menuactive(1);
end;

(*)
(*)           Move Active Menu to Last Menu             (*)
(*)
procedure movetolastmenu;
begin
  windowclose;
  statusmenu:=maxmenu;
  menuactive(maxmenu);
end;

```

```

(*)
(*)          Open Data File Which be Process
(*)
(*)
procedure input_datafile;
var   y,j,io : integer;
      s1,s2,s3 : string[5];
begin
  setwinattr(lowdisplay);
  setboxattr(highdisplay);
  setcharattr(lowdisplay);
  setboxstyle(double);
  setwinheader('Input Data File');
  windowopen(26,10,64,14);
  cursoron;
  repeat
    if fl1 then close(act);
    gotoxy(2,1);write('Enter Number of data : ');
    n:=inputword(3,25,1);
    gotoxy(2,2);write('Enter Number of Y : ');
    y:=inputword(1,22,2);
    gotoxy(2,3);write('Enter Number of J : ');
    j:=inputword(1,22,3);
    Str(n,s1);
    str(y,s2);
    str(j,s3);
    fname:='N'+s1+'Y'+s2+'J'+s3+'.DAT';
    assign(act,fname);
    {$i-} reset(act); {$i+}
    io:=ioresult;
    if io<>0 then begin
      gotoxy(2,4);setattr(reversehigh);
      write('^G,^G,fname,' File not found');
      delay(1000);
      gotoxy(1,4);clreol;
    end;
  until io=0;
  jj:=j;
  yy:=y;
  case n of
    5 : nn:=1;
    10 : nn:=2;
    15 : nn:=3;
    20 : nn:=4;
    30 : nn:=5;
    50 : nn:=6;
    100 : nn:=7;
  end;
  fl1:=true;
  windowclose;
  cursoroff;
  windowclose;
  setattr(reversehigh);
  gotoxy(55,3);write('Data File is ',fname);
  writeln(rpt,'Filename : ',fname);
  setattr(lowdisplay);
  menuactive(1);
  write('^G);
end;

```



```

(*)
(*)      Open Output File Which The Result of Process      (*)
(*)
procedure output_datafile;
var tc : char;
begin
    setwinattr(lowdisplay);
    setboxattr(highdisplay);
    setcharattr(lowdisplay);
    setboxstyle(double);
    setwinheader('Output Data File');
    windowopen(18,10,64,12);
    cursoron;
    if fl2 then close(rpt);
    gotoxy(1,1);write('Enter Output Filename : ');
    fname2:='';
    {$v-}inputstr(fname2,30,25,1,[#13],tc); {$v+}
    assign(rpt,fname2);
    {$i-}reset(rpt); {$i+}
    if ioresult<>0 then rewrite(rpt)
        else append(rpt);
    fl2:=true;
    windowclose;
    cursoroff;
    windowclose;
    setattr(reversehigh);
    gotoxy(55,5);write('Output File is ',fname2);
    setattr(lowdisplay);
    menuactive(1);
    write('^G');
end;
(*)
(*)      Find the Result of A with Power B ( A Power B)      (*)
(*)
function power(a : real;b : integer):real;
var
    x,z : real;
    y : integer;
begin
    x:=a;
    y:=b;
    z:=1;
    while y<>0 do begin
        if odd(y) then begin
            y:=y-1;
            z:=z*x;
        end
        else begin
            y:=y div 2;
            x:=x*x;
        end;
    end;
    end;
    power:=z;
end;

```

```

(*)
(*)           Main Procedure to Call Transform Data           (*)
(*)
(*)
procedure transform_data;
var
  act8,act9 : file of r;
  inf8,inf9 : r;
  f8,f9     : string[50];
  s1,s2,s3  : string[5];
  n1,y,j,io,ik : integer;

(*)
(*) Transform Data Which Y=1, It was use with J In Transform_Data Procedure (*)
(*)
procedure transform1;
var   i : integer;
begin
  for i:=1 to n1 do
    inf9.data[i]:=power(inf8.data[i],j+1);
  end;

(*)
(*) Transform Data Which Y=2, It was use with J In Transform_Data Procedure (*)
(*)
procedure transform2;
var   sum : real;
      i,k : integer;
begin
  for i:=1 to n1 do begin
    randomize;
    sum:=0.0;
    for k:=1 to j+1 do
      sum:=sum+inf8.data[random(n1)+1];
    sum:=sum/(j+1);
    inf9.data[i]:=sum;
  end;
end;

(*)
(*) Transform Data Which Y=3, It was use with J In Transform_Data Procedure (*)
(*)
procedure transform3;
var
  s : real;
  i : integer;
begin
  randomize;
  for i:=1 to n1 do begin
    repeat
      s:=random;
    until s<>0.50;
    if s>0.50 then s:=1.0
    else s:=0.0;
    inf9.data[i]:=((s*power(inf8.data[i],j+1))+
      ((1-s)*(1-power(inf8.data[i],j+1))));
  end;
end;
end;

```

```

(*)
(*)           Main Procedure of Transform_Data Procedure
(*)
(*)
begin
  setwinattr(lowdisplay);
  setboxattr(highdisplay);
  setcharattr(lowdisplay);
  setboxstyle(double);
  setwinheader('Transform Data');
  windowopen(18,10,64,14);
  cursoron;
  repeat
    gotoxy(2,1);write('Enter Number of data : ');
    n1:=inputword(3,25,1);
    gotoxy(2,2);write('Enter Number of Y : ');
    y:=inputword(1,22,2);
    gotoxy(2,3);write('Enter Number of J : ');
    j:=inputword(1,22,3);
    Str(n1,s1);
    str(y,s2);
    str(j,s3);
    if fll then close(act);
    f8:='N'+s1+'Y0J0.DAT';
    f9:='N'+s1+'Y'+s2+'J'+s3+'.DAT';
    assign(act8,f8);
    reset(act8);
    {$i-} reset(act8); {$i+}
    io:=ioresult;
    if io<>0 then begin
      gotoxy(2,4);setattr(reversehigh);
      write('^G,^G,f8,' File not found');
      delay(1000);
      gotoxy(1,4);clrleol;
    end;
  until io=0;
  assign(act9,f9);
  rewrite(act9);
  cursoroff;
  clrscr;
  writeln(' Transform data from ',f8);
  writeln(' ... to ',f9);
  repeat
    read(act8,inf8);
    gotoxy(10,3);
    write('Record # ',filepos(act8));
    case y of
      1 : transform1;
      2 : transform2;
      3 : transform3;
    end;
  end;

```

```

        for ik:=n1+1 to 50 do inf9.data[ik]:=0.0;
        write(act9,inf9);
until eof(act8);
close(act8);
close(act9);
if fl1 then begin
    assign(act,fname);
    reset(act);
end;
windowclose;
write(^G);
end;

(*
(*          Generate Data Which be Process
(*
procedure generate_data;
const    n : array[1..2] of integer=(5,15);
var
    temp    : real;
    i,j,k,l,m : integer;
begin
    setwinattr(lowdisplay);
    setboxattr(highdisplay);
    setcharattr(lowdisplay);
    setboxstyle(double);
    setwinheader('Generate Data');
    windowopen(18,10,64,14);
    for k:=1 to 2 do begin
        str(n[k],fname);
        fname:='N'+fname+'YOJO.DAT';
        gotoxy(2,1);write('Generate data file : ',fname);
        assign(act,fname);
        rewrite(act);
        for j:=1 to 1000 do begin
            gotoxy(2,2);write('Generate at Record # ',j:4);
            randomize;
            for i:=1 to n[k] do inf.data[i]:=random;
            for i:=n[k]+1 to 50 do inf.data[i]:=0.0;
            for l:=1 to n[k]-1 do begin
                for m:=l+1 to n[k] do begin
                    if inf.data[l]>inf.data[m] then begin
                        temp:=inf.data[l];
                        inf.data[l]:=inf.data[m];
                        inf.data[m]:=temp;
                    end;
                end;
            end;
            write(act,inf);
        end;
        close(act);
    end;
    windowclose;
    write(^G);
end;

```



```

close(t);
setwinattr(lowdisplay);
setboxattr(highdisplay);
setcharattr(lowdisplay);
setboxstyle(double);
setwinheader('Print Report');
windowopen(18,10,50,12);
gotoxy(2,1);write('^g','Report are in file REPORT.RPT');
repeat until keypressed;
windowclose;

end;
(*
(*           Test for Uniformity or not
(*
(*
procedure test_uniform;
var
  act : file of r;
  inf,inf2 : r;
  fname,fname2 : string[50];
  rpt : text;
  n : integer;
  y,j,count,count1 : integer;
  nn,yy,jj : string[3];
  ax,ay : array[0..12] of boolean;
  cx,cy : array[0..12] of real;
  ch,cch : char;

procedure transform1(inf8:r ; var inf9 : r; y,j,n : integer);
var
  i : integer;
  inftmp : r;
begin
  for i:=1 to n do
    inftmp.data[i]:=power(inf8.data[i],j+1);
  inf9:=inftmp;
end;

procedure transform2(inf8:r ; var inf9 : r; y,j,n : integer);
var
  sum : real;
  i,k : integer;
  inftmp : r;
begin
  randomize;
  for i:=1 to n do begin
    sum:=0.0;
    for k:=1 to j+1 do
      sum:=sum+inf8.data[random(n)+1];
    sum:=sum/(j+1);
    inftmp.data[i]:=sum;
  end;
  inf9:=inftmp;
end;

```

```

procedure transform3(inf8:r ; var inf9 : r; y,j,n : integer);
var
    s : real;
    i : integer;
    inftmp : r;
begin
    randomize;
    for i:=1 to n do begin
        repeat
            s:=random;
        until s<>0.50;
        if s>0.50 then s:=1.0
        else s:=0.0;
        inftmp.data[i]:= (s*power(inf8.data[i],j+1))+
            ((1-s)*(1-power(inf8.data[i],j+1)));
        end;
        inf9:=inftmp;
    end;

procedure chi_square(inf : r;n : integer;var chail : real);
var
    table : array[1..8,1..2] of real;
    i,j : integer;
    chi : real;
begin
    for i:=1 to 8 do
        for j:=1 to 2 do table[i,j]:=0.0;
    for i:=1 to n do begin
        case trunc(inf.data[i]*1e4) of
            0000..1250 : table[1,1]:=table[1,1]+1;
            1251..2500 : table[2,1]:=table[2,1]+1;
            2501..3750 : table[3,1]:=table[3,1]+1;
            3751..5000 : table[4,1]:=table[4,1]+1;
            5001..6250 : table[5,1]:=table[5,1]+1;
            6251..7500 : table[6,1]:=table[6,1]+1;
            7501..8750 : table[7,1]:=table[7,1]+1;
            8751..10000 : table[8,1]:=table[8,1]+1;
        end;
    end;
    for i:=1 to 8 do table[i,2]:=table[i,1]*(1/8);
    chi :=0.0;
    for i:=1 to 8 do
        if table[i,2]>0 then
            chi:=chi+(Sqr(table[i,1]-n/8)/(n/8));
        chil:=chi;
    end;
end;

```

```
clrscr;
gotoxy(20,1);write('Enter N : ');
n:=inputword(3,30,1);
str(n,nn);
fname:='N'+nn+'Y0J0.DAT';
assign(act,fname);
reset(act);
assign(rpt,'Test.XXX');
rewrite(rpt);
count:=0;
repeat
  gotoxy(1,1);write('Record : ',filepos(act):3);
  count1:=0;
  for n:=0 to 12 do begin
    ax[n]:=false;
    cx[n]:=0.0;
  end;
  read(act,inf);
  chai_square(inf,n,cx[0]);
  if cx[0]<=14.10 then begin
    ax[0]:=true;
    count1:=count1+1;
  end
  else ax[0]:=false;
  for y:=1 to 3 do begin
    for j:=1 to 4 do begin
      case y of
        1 : transform1(inf,inf2,y,j,n);
        2 : transform2(inf,inf2,y,j,n);
        3 : transform3(inf,inf2,y,j,n);
      end;
      chi_square(inf2,n,cx[(y-1)*4+j]);
      if cx[(y-1)*4+j]<=14.10 then begin
        ax[(y-1)*4+j]:=true;
        count1:=count1+1;
      end
      else ax[(y-1)*4+j]:=false;
    end;
  end;
end;
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย


```

if count1>count then begin
  ay:=ax;
  cy:=cx;
  count:=count1;
  gotoxy(15,20);write(count1:5,count:5);
  writeln(rpt,'Chi-Square');
  gotoxy(10,2);write(filepos(act)-1:3);
  for y:=0 to 12 do begin
    gotoxy(1,y+3);
    write(ay[y]:5,' ');
    write(rpt,ay[y]:5,' '); end;
  writeln(rpt);
  for y:=0 to 12 do begin
    gotoxy(11,y+3);
    write(cy[y]:5:4,' ');
    write(rpt,cy[y]:5:4,' ');
  end;
  writeln(rpt);
  writeln(rpt);
end;
gotoxy(15,20);write(count1:5,count:5);
for y:=0 to 12 do begin
  gotoxy(51,y+3);
  write(ax[y]:5,' ',cx[y]:6:4);
end;

if keypressed then begin
  cch:=readkey;
end;
until (eof(act)) or (cch=#27);
close(act);
close(rpt);
end;

```

ศูนย์วิทยพัชกร
จุฬาลงกรณ์มหาวิทยาลัย

```

(*)
(*)          Separate to Call of Choice in Menu # 1
(*)
procedure domenu1;
begin
  case choicemenu[statusmenu] of
    1 : generate_data;
    2 : transform_data;
    3 : input_datafile;
    4 : output_datafile;
    5 : printreport;
    6 : Test_uniform;
  end;
end;

(*)
(*)          Compute Pearson Chi Square Test 8 Level
(*)
procedure chi_square(n : integer; var f1, f2 : boolean);
var
  table : array[1..8, 1..2] of real;
  i, j : integer;
  chi : real;
begin
  for i:=1 to 8 do
    for j:=1 to 2 do table[i, j]:=0.0;
  end;
  for i:=1 to n do begin
    case trunc(inf.data[i]*1e4) of
      0000..1250 : table[1, 1]:=table[1, 1]+1;
      1251..2500 : table[2, 1]:=table[2, 1]+1;
      2501..3750 : table[3, 1]:=table[3, 1]+1;
      3751..5000 : table[4, 1]:=table[4, 1]+1;
      5001..6250 : table[5, 1]:=table[5, 1]+1;
      6251..7500 : table[6, 1]:=table[6, 1]+1;
      7501..8750 : table[7, 1]:=table[7, 1]+1;
      8751..10000 : table[8, 1]:=table[8, 1]+1;
    end;
  end;
  for i:=1 to 8 do table[i, 2]:=table[i, 1]*(1/8);
  chi :=0.0;
  for i:=1 to 8 do
    if table[i, 2]>0 then
      chi:=chi+(Sqr(table[i, 1]-n/8)/(n/8));
    end;
  end;
  if chi>18.50 then f1:=true
  else f1:=false;
  if chi>14.10 then f2:=true
  else f2:=false;
end;

```

```

(*)
(*)          Compute Pearson Chi Square Test 5 Level
(*)
procedure chi_square1(n : integer; var f1, f2 : boolean);
var
  table : array[1..5, 1..2] of real;
  i, j  : integer;
  chi   : real;
begin
  for i:=1 to 5 do
    for j:=1 to 2 do table[i, j]:=0.0;
  for i:=1 to n do begin
    case trunc(inf.data[i]*1e4) of
      0000..2000 : table[1, 1]:=table[1, 1]+1;
      2001..4000 : table[2, 1]:=table[2, 1]+1;
      4001..6000 : table[3, 1]:=table[3, 1]+1;
      6001..8000 : table[4, 1]:=table[4, 1]+1;
      8001..10000 : table[5, 1]:=table[5, 1]+1;
    end;
  end;
  for i:=1 to 5 do table[i, 2]:=table[i, 1]*(1/5);
  chai:=0.0;
  for i:=1 to 5 do
    if table[i, 2]>0 then
      chi:=chi+(Sqr(table[i, 1]-n/5)/(n/5));
  if chi>13.30 then f1:=true
  else f1:=false;
  if chi>9.49 then f2:=true
  else f2:=false;
end;

(*)
(*)          Compute Kolmogorov Smirnov Test
(*)
procedure test_d(n : integer; var f1, f2 : boolean);
var i : integer;
    max, td, t1, t2 : real;
begin
  Max:=0.0;
  for i:=1 to n do begin
    t1:=abs(inf.data[i]-(i-1)/n);
    t2:=abs(inf.data[i]-i/n);
    if t1>t2 then begin
      if t1>max then max:=t1;
    end
    else if t2>max then max:=t2;
  end;
  td := Max* (Sqr(n)+0.12+0.11 /Sqr(n));
  if td >1.628 then f1:=true
  else f1:=false;
  if td >1.358 then f2:=true
  else f2:=false;
end;

```

```

(*)
(*)          Compute Watson Test          (*)
(*)
(*)
procedure test_U_square(n : integer; var f1, f2 : boolean);
var
  mean, tu, sum, u : real;
  i                : integer;
begin
  mean:=0.0; sum:=0.0;
  for i:=1 to n do begin
    mean:=mean+inf.data[i];
    sum:=sum+Sqr(((2*i-1)/(2*n))-inf.data[i]);
  end;
  mean:=mean/n;
  u:=1/(12*n)+sum-(n*Sqr(mean-0.5));
  tu:=(u-(0.1/n)+(0.1/Sqr(n)))*(1.0+(0.8/n));
  if tu > 0.267 then f1:=true
    else f1:=false;
  if tu > 0.187 then f2:=true
    else f2:=false;
end;

(*)
(*)          Compute Anderson Darling Test (*)
(*)
(*)
procedure test_A(n : integer; var f1, f2 : boolean);
var
  i, j : integer;
  sum , a, b, c, d : real;
begin
  sum:=0.0;
  for i:=1 to n do begin
    if inf.data[i]=0.0 then b:=0.0
    else b:=ln(inf.data[i]);
    if inf.data[n+1-i]=1.0 then c:=0
    else c:=ln(1-inf.data[n+1-i]);
    d:=b+c;
    sum:=sum+(2*i-1)*d;
  end;
  a:=-(1/n)*sum-n;
  if a > 3.857 then f1:=true
    else f1:=false;
  if a > 2.492 then f2:=true
    else f2:=false;
end;

```

```

(*)
(*)           Compute Q Statistic Test
(*)
procedure test_Q(n : integer; var f1, f2 : boolean);
var i : integer;
    sum1, sum2, q, t1, t2 : real;
    ta : array[0..101] of real;
begin
  for i:=1 to n do ta[i]:=inf.data[i];
  ta[0]:=0.0;
  ta[n+1]:=1.0;
  sum1:=0.0;
  for i:=1 to n+1 do
    sum1:=sum1+Sqr(ta[i]-ta[i-1]);
  sum2:=0.0;
  for i:=1 to n do
    sum2:=sum2+((ta[i]-ta[i-1])*(ta[i+1]-ta[i]));
  q:=sum1+sum2;
  case n of
    5 : begin t1:=0.642;t2:=0.551; end;
    10 : begin t1:=0.378;t2:=0.319; end;
    15 : begin t1:=0.259;t2:=0.222; end;
    20 : begin t1:=0.196;t2:=0.168; end;
    30 : begin t1:=0.130;t2:=0.114; end;
    50 : begin t1:=0.075;t2:=0.068; end;
    100 : begin t1:=0.036;t2:=0.033; end;
  end;
  if q > t1 then f1:=true
  else f1:=false;
  if q > t2 then f2:=true
  else f2:=false;
end;

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

(*)
(*)          Compute Mean, Standard Deviation and CV          (*)
(*)
procedure mean_and_SD(n: integer);
var  me      : real;
     mean    : array[1..1000] of real;
     rmean   : real;
     sd      : real;
     sum     : real;
     i,j     : integer;
     ch      : char;
     cv      : real;
begin
  reset(act);
  j:=1;
  sum:=0;
  repeat
    read(act,inf);
    for i:=1 to n do sum:=sum+inf.data[i];
  until eof(act);
  rmean:=sum/(n*1000);
  reset(act);
  sum:=0.0;
  repeat
    read(act,inf);
    for i:=1 to n do sum:=sum+sqr(inf.data[i]-rmean);
  until eof(act);
  sd:=sqrt(sum/((n*1000)-1));
  cv:=sd/rmean;
  gotoxy(1,1);
  write('^G,'Mean of population = ',rmean:8:6);clreol;
  gotoxy(1,2);
  write('SD of population = ',sd:8:6);clreol;
  gotoxy(1,3);
  write('CV of population = ',cv:8:6);clreol;
  writeln(rpt,'          Mean of population is ',rmean:8:6);
  writeln(rpt,'          SD of population is ',sd:8:6);
  writeln(rpt,'          CV of population is ',CV:8:6);
  writeln(rpt);
  inf1.p[1,nn,yy,jj,7]:=rmean;
  inf1.p[1,nn,yy,jj,8]:=sd;
  inf1.p[1,nn,yy,jj,9]:=cv;
end;

```

```

(*)
(*)          Separate to Call of Choice in Menu # 2
(*)
procedure domenu2;
var
    n_reject1,n_reject2 : integer;
    fl,f2      : boolean;
begin
    setwinattr(lowdisplay);
    setboxattr(highdisplay);
    setcharattr(lowdisplay);
    setboxstyle(double);
    xx:=choicemenu(statusmenu);
    case choicemenu[statusmenu] of
        1 : begin setwinheader('Chi Square Test 8 level');
                writeln(rpt,'=====> Pearson Chi Square Test 8 Level');
            end;
        2 : begin setwinheader('Chi Square Test 5 level');
                writeln(rpt,'=====> Pearson Chi Square Test 5 Level');
            end;
        3 : begin setwinheader('Kolmogorov Smirnov Test');
                writeln(rpt,'=====> Kolmogorov Smirnov Test');
            end;
        4 : begin setwinheader('Watson Test');
                writeln(rpt,'=====> Watson Test');
            end;
        5 : begin setwinheader('Anderson Darling Test');
                writeln(rpt,'=====> Anderson Darling Test');
            end;
        6 : begin setwinheader('Q Statistic Test');
                writeln(rpt,'=====> Q Statistic Test');
            end;
        7 : begin setwinheader('Mean & SD');
                writeln(rpt,'=====> Mean and SD = ');
            end;
    end;
    windowopen(18,10,70,14);
    if not(fl1) then begin
        gotoxy(2,1);
        write('^g,^g, 'File does not Open');
    end;
    delay(1000);
    windowclose;
    exit;
end;
setattr(revblinklow);
gotoxy(7,2);write('Process in Progress... Please Wait');
setattr(lowdisplay);
reset(act);
n_reject1:=0;
n_reject2:=0;

```

```

if choicemenu[statusmenu] in [1..6] then begin
  repeat
    read(act,inf);
    case choicemenu[statusmenu] of
      1 : chi_square(n,f1,f2);
      2 : chi_square1(n,f1,f2);
      3 : test_d(n,f1,f2);
      4 : test_U_square(n,f1,f2);
      5 : test_A(n,f1,f2);
      6 : test_Q(n,f1,f2);
    end;
    if choicemenu[statusmenu] in [1..6] then begin
      if f1 then n_reject1:=n_reject1+1;
      if f2 then n_reject2:=n_reject2+1;
    end;
  until eof(act);
  if choicemenu[statusmenu] in [1..6] then begin
    gotoxy(1,1);
    write('^G,'Number of reject at 0.01 is ',n_reject1:4);clreol;
    gotoxy(1,2);
    write('Number of reject at 0.05 is ',n_reject2:4);clreol;
    writeln(rpt,'      Number of reject at 0.01 is ',n_reject1:4);
    writeln(rpt,'      Number of reject at 0.05 is ',n_reject2:4);
    writeln(rpt);
    inf1.p[1,nn,yy,jj,xx]:=n_reject1/1000;
    inf1.p[2,nn,yy,jj,xx]:=n_reject2/1000;
  end;
end
else begin
  mean_And_sd(n);
end;
repeat until keypressed;
windowclose;
end;

(*
(*          Separate to Call of Menu
(*
procedure domenu;
begin
  case statusmenu of
    1 : domenu1;
    2 : domenu2;
  end;
end;

```



```
(*                                     *)
(*                                     *)
(*                                     *)
procedure quit;
begin
    setattr(highdisplay);
    windowclose;
    cursoron;
    clrscr;
    finish:=true;
end;

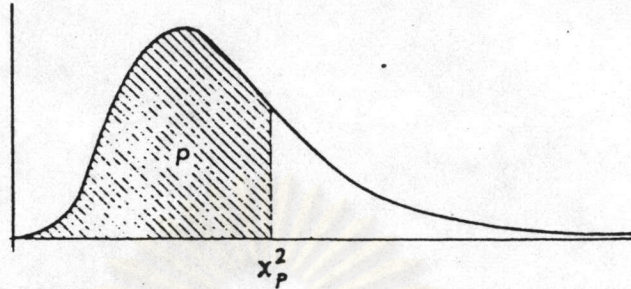
(*                                     *)
(*                                     *)
(*                                     *)
procedure testkey(key : char);
begin
    if funckey then
        case key of
            home_key : movetofirstmenu;
            end_key   : movetolastmenu;
            Up_key    : moveup;
            Lt_key    : moveback;
            Rt_key    : moveforward;
            Dn_key    : movedown;
            Pgup_key  : movetofirstchoice;
            Pgdn_key  : movetolastchoice;
            F10_key   : quit;
        end
    else if key =Return_key then donenu;
end;
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
(* *)
(* Main Program *)
(* *)
begin
  clrscr;
  directvideo:=true;
  checksnow :=true;
  displayfunkey;
  displaymainmenu;
  menuactive(1);
  cursoroff;
  fl1:=false;
  fl2:=false;
  assign(axt,'res.ref');
  reset(axt);
  read(axt,infl);
  repeat
    readfunkey(key);
    testkey(key);
  until finish;
  if fl1 then close(act);
  if fl2 then close(rpt);
  reset(axt);
  write(axt,infl);
  close(axt);
end.
(* *)
(* End Main Program *)
(* *)
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

TABLE I Percentiles of the Chi-Square Distribution.



Degrees of freedom	$\chi^2_{.005}$	$\chi^2_{.01}$	$\chi^2_{.025}$	$\chi^2_{.05}$	$\chi^2_{.10}$	$\chi^2_{.20}$	$\chi^2_{.30}$	$\chi^2_{.50}$	$\chi^2_{.70}$	$\chi^2_{.80}$	$\chi^2_{.90}$	$\chi^2_{.95}$	$\chi^2_{.975}$	$\chi^2_{.99}$	$\chi^2_{.995}$
1	.000	.000	.001	.004	.016	.064	.148	.455	1.07	1.64	2.71	3.84	5.02	6.63	7.88
2	.010	.020	.051	.103	.211	.446	.713	1.39	2.41	3.22	4.61	5.99	7.38	9.21	10.6
3	.072	.115	.216	.352	.584	1.00	1.42	2.37	3.66	4.64	6.25	7.81	9.35	11.3	12.8
4	.207	.297	.484	.711	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	11.1	13.3	14.9
5	.412	.554	.831	1.15	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.1	12.8	15.1	16.7
6	.676	.872	1.24	1.64	2.20	3.07	3.83	5.35	7.23	8.56	10.6	12.6	14.4	16.8	18.5
7	.989	1.24	1.69	2.17	2.83	3.82	4.67	6.35	8.38	9.80	12.0	14.1	16.0	18.5	20.3
8	1.34	1.65	2.18	2.73	3.49	4.59	5.53	7.34	9.52	11.0	13.4	15.5	17.5	20.1	22.0
9	1.73	2.09	2.70	3.33	4.17	5.38	6.39	8.34	10.7	12.2	14.7	16.9	19.0	21.7	23.6
10	2.16	2.56	3.25	3.94	4.87	6.18	7.27	9.34	11.8	13.4	16.0	18.3	20.5	23.2	25.2
11	2.60	3.05	3.82	4.57	5.58	6.99	8.15	10.3	12.9	14.6	17.3	19.7	21.9	24.7	26.8
12	3.07	3.57	4.40	5.23	6.30	7.81	9.03	11.3	14.0	15.8	18.5	21.0	23.3	26.2	28.3
13	3.57	4.11	5.01	5.89	7.04	8.63	9.93	12.3	15.1	17.0	19.8	22.4	24.7	27.7	29.8
14	4.07	4.66	5.63	6.57	7.79	9.47	10.8	13.3	16.2	18.2	21.1	23.7	26.1	29.1	31.3
15	4.60	5.23	6.26	7.26	8.55	10.3	11.7	14.3	17.3	19.3	22.3	25.0	27.5	30.6	32.8
16	5.14	5.81	6.91	7.96	9.31	11.2	12.6	15.3	18.4	20.5	23.5	26.3	28.8	32.0	34.3
17	5.70	6.41	7.56	8.67	10.1	12.0	13.5	16.3	19.5	21.6	24.8	27.6	30.2	33.4	35.7
18	6.26	7.01	8.23	9.39	10.9	12.9	14.4	17.3	20.6	22.8	26.0	28.9	31.5	34.8	37.2
19	6.83	7.63	8.91	10.1	11.7	13.7	15.4	18.3	21.7	23.9	27.2	30.1	32.9	36.2	38.6
20	7.43	8.26	9.59	10.9	12.4	14.6	16.3	19.3	22.8	25.0	28.4	31.4	34.2	37.6	40.0
21	8.03	8.90	10.3	11.6	13.2	15.4	17.2	20.3	23.9	26.2	29.6	32.7	35.5	38.9	41.4
22	8.64	9.54	11.0	12.3	14.0	16.3	18.1	21.3	24.9	27.3	30.8	33.9	36.8	40.3	42.8
23	9.26	10.2	11.7	13.1	14.8	17.2	19.0	22.3	26.0	28.4	32.0	35.2	38.1	41.6	44.2
24	9.89	10.9	12.4	13.8	15.7	18.1	19.9	23.3	27.1	29.6	33.2	36.4	39.4	43.0	45.6
25	10.5	11.5	13.1	14.6	16.5	18.9	20.9	24.3	28.2	30.7	34.4	37.7	40.6	44.3	46.9
26	11.2	12.2	13.8	15.4	17.3	19.8	21.8	25.3	29.2	31.8	35.6	38.9	41.9	45.6	48.3
27	11.8	12.9	14.6	16.2	18.1	20.7	22.7	26.3	30.3	32.9	36.7	40.1	43.2	47.0	49.6
28	12.5	13.6	15.3	16.9	18.9	21.6	23.6	27.3	31.4	34.0	37.9	41.3	44.5	48.3	51.0
29	13.1	14.3	16.0	17.7	19.8	22.5	24.6	28.3	32.5	35.1	39.1	42.6	45.7	49.6	52.3
30	13.8	15.0	16.8	18.5	20.6	23.4	25.5	29.3	33.5	36.2	40.3	43.8	47.0	50.9	53.7
40	20.7	22.1	24.4	26.5	29.0	32.3	34.9	39.3	44.2	47.3	51.8	55.8	59.3	63.7	66.8
50	28.0	29.7	32.3	34.8	37.7	41.3	44.3	49.3	54.7	58.2	63.2	67.5	71.4	76.2	79.5
60	35.5	37.5	40.5	43.2	46.5	50.6	53.8	59.3	65.2	69.0	74.4	79.1	83.3	88.4	92.0

Note: For degrees of freedom $k > 30$, use $\chi_p^2 = \frac{1}{2}(z_p + \sqrt{2k-1})^2$, where z_p is the corresponding percentile of the standard normal distribution.

This table is adapted from Table VIII of *Biometrika Tables for Statisticians*, Vol. 1, 1954, by E. S. Pearson and H. O. Hartley, originally prepared by Catherine M. Thompson, with the kind permission of the editor of *Biometrika*.

TABLE II
Percentage points of Q statistic

N	α					N	α				
	0.50	0.10	0.05	0.01	0.001		0.50	0.10	0.05	0.01	0.001
2	0.659	0.811	0.859	0.932	0.977	31	0.088	0.104	0.110	0.125	0.148
3	0.527	0.691	0.736	0.831	0.920	32	0.086	0.101	0.106	0.120	0.142
4	0.447	0.586	0.635	0.727	0.829	33	0.083	0.097	0.103	0.116	0.135
5	0.388	0.505	0.551	0.642	0.739	34	0.081	0.095	0.100	0.112	0.131
6	0.343	0.442	0.483	0.573	0.691	35	0.079	0.092	0.097	0.110	0.128
7	0.307	0.393	0.429	0.512	0.622	36	0.077	0.090	0.095	0.107	0.124
8	0.278	0.355	0.387	0.463	0.551	37	0.075	0.087	0.092	0.103	0.122
9	0.254	0.322	0.350	0.423	0.506	38	0.073	0.085	0.089	0.100	0.118
10	0.234	0.294	0.319	0.378	0.461	39	0.071	0.083	0.087	0.097	0.114
11	0.217	0.272	0.294	0.351	0.434	40	0.070	0.081	0.085	0.095	0.112
12	0.202	0.251	0.272	0.318	0.392	41	0.068	0.079	0.083	0.092	0.110
13	0.189	0.234	0.253	0.298	0.371	42	0.067	0.077	0.081	0.090	0.107
14	0.177	0.220	0.237	0.279	0.348	43	0.065	0.075	0.079	0.088	0.103
15	0.168	0.206	0.222	0.259	0.321	44	0.064	0.073	0.077	0.086	0.101
16	0.159	0.195	0.209	0.245	0.294	45	0.062	0.072	0.075	0.084	0.098
17	0.150	0.184	0.197	0.230	0.278	46	0.061	0.070	0.074	0.082	0.094
18	0.143	0.174	0.187	0.218	0.268	47	0.060	0.069	0.072	0.080	0.092
19	0.137	0.166	0.177	0.206	0.257	48	0.059	0.067	0.070	0.078	0.091
20	0.131	0.158	0.168	0.196	0.246	49	0.058	0.066	0.069	0.077	0.090
21	0.125	0.151	0.162	0.187	0.232	50	0.057	0.065	0.068	0.075	0.086
22	0.120	0.144	0.154	0.178	0.218	55	0.052	0.059	0.061	0.068	0.078
23	0.115	0.138	0.147	0.169	0.206	60	0.048	0.054	0.056	0.062	0.070
24	0.111	0.133	0.141	0.163	0.192	65	0.044	0.050	0.052	0.057	0.064
25	0.107	0.128	0.136	0.156	0.189	70	0.041	0.046	0.048	0.052	0.059
26	0.103	0.123	0.131	0.148	0.182	75	0.038	0.043	0.045	0.048	0.054
27	0.100	0.119	0.126	0.144	0.173	80	0.036	0.040	0.042	0.045	0.051
28	0.097	0.114	0.121	0.138	0.166	85	0.034	0.038	0.039	0.042	0.047
29	0.094	0.111	0.117	0.134	0.158	90	0.032	0.036	0.037	0.040	0.044
30	0.091	0.107	0.114	0.130	0.155	95	0.031	0.034	0.035	0.038	0.041
						100	0.029	0.032	0.033	0.036	0.040

Table III. Modifications yielding approximate percentage points for the statistics D , V , W^2 , U^2 and A in finite samples of n observations

Case 1: Modifications for the test when $F(x)$ completely known

Statistic	Modified forms $T(D^+)$, $T(D)$, $T(V)$, etc.	Upper percentage points for modified T				
		15.0	10.0	5.0	2.5	1.0
D^+ (D^-)	$D^+(\sqrt{n} + 0.12 + 0.11/\sqrt{n})$	0.973	1.073	1.224	1.358	1.518
D	$D(\sqrt{n} + 0.12 + 0.11/\sqrt{n})$	1.138	1.224	1.358	1.480	1.628
V	$V(\sqrt{n} + 0.155 + 0.24/\sqrt{n})$	1.537	1.620	1.747	1.862	2.001
W^2	$(W^2 - 0.4/n + 0.6/n^2)(1.0 + 1.0/n)$	0.284	0.347	0.461	0.581	0.743
U^2	$(U^2 - 0.1/n + 0.1/n^2)(1.0 + 0.8/n)$	0.131	0.152	0.187	0.221	0.267
A	For all $n \geq 5$:*	1.61	1.933	2.492	3.020	3.857

* Marshall (1958) showed that the distribution of A for $n = 1$ gives results remarkably close to the asymptotic ones, at least in the upper tail. To fill the gap Stephens carried out a Monte Carlo study for $n = 5$ and found the upper tail asymptotic points to be very close to the corresponding Monte Carlo points.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ประวัติผู้เขียน

นางสาวยุพาพร รักศิลป์เกิด วันที่ 15 กุมภาพันธ์ พ.ศ. 2501 จังหวัด กรุงเทพมหานคร ได้รับปริญญาวิทยาศาสตรบัณฑิต (สเกิต) มหาวิทยาลัยเชียงใหม่ ปีการศึกษา 2525 และได้เข้าศึกษาต่อในสาขาวิชาสถิติ ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2528 ขณะดำรงตำแหน่งอาจารย์ประจำคณะวิทยาศาสตร์ สาขาวิชาสถิติ มหาวิทยาลัยพายัพ จังหวัดเชียงใหม่.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย