

ภาคผนวก ค.

โปรแกรมการคำนวณความแปรปรวนของตัวแปรค่าสอดคล้องประชากร T_1 กับ T_2 เมื่อทุกหน่วยของประชากรมีความน่าจะเป็นที่จะให้ข้อมูลเท่ากัน

```

var loop1,loop2,pop,n,m,i,j : longint;
    mean,sd : real;
    p,p0,pi_ij,pi_i,v1,v2,v3 : real;
    data : array [1..1000] of real;
    data_n : array [1..1000] of real;
    data_m : array [1..1000] of real;
    check,ch : array [1..1000] of byte;
    f : text;

function normal : real;
var sum1 : real;
    l : longint;
begin
    sum1 := 0;
    for l := 1 to 12 do
        sum1 := sum1 + random;
    sum1 := sum1 - 6;
    normal := mean + sum1*sd;
end;

procedure make_pop;
var i : longint;

```

```
begin
  for i := 1 to pop do
    data[i] := normal;
  end;
end;
```

```
procedure sampling_n;
var i,j : longint;
begin
  for i := 1 to pop do
    check[i] := 0;
  for i := 1 to n do
    begin
      repeat
        j := random(pop)+1;
      until check[j] = 0;
      data_n[i] := data[j];
      check[j] := 1;
    end;
  end;
end;
```

```
procedure sampling_m;
var i,j : longint;
begin
  for i := 1 to n do
    check[i] := 0;
  for i := 1 to m do
    begin
```

```

repeat
  j := random(n)+1;
until check[j] = 0;
data_m[i] := data_n[j];
check[j] := 1;
end;
end;

function v_t_1 : real;
var i,j,k : longint;
    s1,s2,s3 : real;
begin
  s1 := 0;
  for i := 1 to m do
    s1 := s1 + data_m[i]*data_m[i];
  s3 := s1 * pop * pop *(1-p)/(n * n * p * p);
  s1 := s1 / p;
  s2 := 0;
  for i := 1 to m-1 do
    for j := i+1 to m do
      s2 := s2 + data_m[i] * data_m[j];
    s2 := s2 * 2 / (p*p);
  s2 := s1 - (s1 + s2)/n;
  v_t_1 := pop * (pop - n) / (n * (n-1)) * s2 + s3;
end;

function v_t_2 : real;

```

```

var i,j,k : longint;
    s1,s2,s3 : real;

begin
    s1 := 0;

    s2 := 0;

    s3 := 0;

    for i := 1 to m-1 do
        for j := i+1 to m do
            begin
                s1 := s1 + data_m[i] * data_m[j];
                s2 := s2 + (data_m[i]-data_m[j]) * (data_m[i]-data_m[j]);
            end;
        s1 := 2 * s1 / pi_ij;
        s2 := s2*(1/pi_ij - 1/(pi_i*pi_i));
        s2 := s2 * pop * pop / (n * n * (1-p0) * (1-p0));
        for i := 1 to m do
            s3 := s3 + data_m[i] * data_m[i];
        s3 := s3 / pi_i;
        s1 := s3 - (s3 + s1)/n;
        s1 := s1 * pop * (pop-n) / (n*(n-1));
        v_t_2 := s1 + s2;
    end;

function v_t_3 : real;
var i,j,k : longint;
    s1,s2,s3 : real;

begin
    s3 := 0;

```

```

for i := 1 to m-1 do
  for j := i+1 to m do
    s3 := s3 + data_m[i] * data_m[j];
s3 := s3 * 2 / pi_ij;
s1 := 0;
for i := 1 to m do
  s1 := s1 + data_m[i];
s1 := s1 / pi_i ;
s1 := s1 + s3;
s1 := s1 / (n-1);
s2 := 0;
for i := 1 to m do
  s2 := s2 + data_m[i] * data_m[i];
s2 := s2 / pi_i;
s3 := s2 - (s3 + s2)/n;
s2 := pop * (pop - n)/(n * (n-1)) * s3;
s1 := s3/(n-1);
s1 := s2 + pop * pop * (n * p * (1-p) + (1-p))/(n*n*p*p)*s1;
v_t_3 := s1;
end;
procedure read_data;
begin
  repeat
    write('N : ');
    readln(pop);
  until pop > 0;
  repeat

```

```
    write('n : ');
    readln(n);
until n <= pop ;
write('Mean : ');
readln(mean);
repeat
    write('Sd : ');
    readln(sd);
until sd >= 0;
repeat
    write('Loop1 : ');
    readln(loop1);
until loop1 > 0;
repeat
    write('Loop2 : ');
    readln(loop2);
until loop2 > 0;
end;
procedure value;
var i : longint;
begin
    p := m/n;
    pi_i := m/n;
    pi_ij := m*(m-1)/(n*(n-1));
    p0 := 1-p;
    for i := 1 to n-1 do
        begin
```

```

    p0 := p0 * (1-p);
end;
end;
begin
    randomize;
    assign(f,'result');
    rewrite(f);
    read_data;
    writeln(f,'N = ',pop);
    writeln(f,'n = ',n);
    writeln(f,'Mean = ',mean:7:3);
    writeln(f,'Sd = ',sd:7:3);
    writeln(f);
    writeln('n = ',n);
    writeln('Mean = ',mean:7:3);
    writeln('Sd = ',sd:7:3);
    writeln;
    make_pop;
    sampling_n;
    for i := 1 to n do
    begin
        write(data_nfil:10:2);
        write(f,data_nfil:10:2);
    end;
    writeln;
    writeln(f);
    writeln(f,' m      rep          V(T1)          V(T2)          V(T3)

```

```
writeln(' m rep          V(T1)          V(T2)          V(T3)
for i := 1 to 1000 do
begin
  ch[i] := 0;
end;
for i := 1 to loop1 do
begin
  writeln(f);
  writeln;
  repeat
    m := random(n div 2)+(n div 2);
  until ch[m] = 0;
  write(f,m:3);
  write(m:3);
  value;
  ch[m] :=1;
  for j := 1 to loop2 do
  begin
    sampling_m;
    v1 := v_t_1;
    v2 := v_t_2;
    v3 := v_t_3;
    if j = 1 then
    begin
      write(f,j:4);
      write(j:4);
    end else
```



```
begin
    write(f,j:7);
    write(j:7);
end;
writeln(f,'      ',v1:18:3,v2:18:3,v3:18:3,v1/v2:10:3);
writeln('      ',v1:18:3,v2:18:3,v3:18:3,v1/v2:10:3);
end;
end;
close(f);
end.
```

THANMASAT UNIVERSITY

สำนักหอสมุด

ตัวอย่าง การใช้โปรแกรมคำนวณความแปรปรวนของตัวประมาณค่ายอดรวมประชากร T_1 กับ T_2 เมื่อแต่ละหน่วยในประชากรให้ข้อมูลด้วยความน่าจะเป็นเท่ากัน

ต้องการศึกษาค่าประมาณยอดรวมประชากร จากตัวอย่างขนาด 10 ซึ่งสุ่มมาโดยไม่ใส่คืน จากประชากรขนาด 20 ที่มีการแจกแจงแบบปกติ ค่าเฉลี่ย 14.650 และความแปรปรวน 10.5625 มีรายละเอียดดังนี้

ขั้นที่ 1 สุ่มประชากรขนาด 20 ที่มีการแจกแจงแบบปกติ โดยมีค่าเฉลี่ย 14.65 ความแปรปรวน 10.5625 ได้ข้อมูลดังนี้

14.97	15.27	17.40	13.75	16.78	19.73	13.94	15.73	13.71	10.64
15.39	12.66	16.90	19.42	13.68	18.57	17.31	15.27	16.91	14.92

ค่ายอดรวมประชากร คือ 312.93

ขั้นที่ 2 สุ่มตัวอย่างขนาด 10 จากประชากรในขั้นที่ 1. ได้ตัวอย่างดังนี้

18.57	15.36	19.42	15.73	14.797	19.73	12.66	10.64	13.71	13.94
-------	-------	-------	-------	--------	-------	-------	-------	-------	-------

ขั้นที่ 3 จากตัวอย่างในขั้นที่ 2. สมมุติว่ามีจำนวนที่ให้ข้อมูล (m) = 8 ได้แก่

13.94	10.64	14.97	15.73	19.73	12.66	15.57	19.42
-------	-------	-------	-------	-------	-------	-------	-------

ขั้นที่ 4 เนื่องจากเป็นการเลือกตัวอย่างแบบ binomial sampling ดังนั้นจะได้ว่า $p = m/n = 8/10 = 0.8$ โดยที่ $p_i = p$ ($i = 1, 2, \dots, N$) = 0.8

ขั้นที่ 5 คำนวณค่าประมาณยอดรวมประชากร $T_1 = \frac{N}{n} \sum_{i=1}^m y_i$

$$= 314.137$$

$$\begin{aligned} \text{คำนวณ } \hat{V}(T_1) &= \frac{N(N-n)}{n(n-1)} \left[\sum_{i=1}^m \frac{y_i^2}{p_i} - \frac{1}{n} \left(\sum_{i=1}^m \frac{y_i^2}{p_i} + \sum_{i \neq j} \frac{y_i y_j}{p_i p_j} \right) \right] + \\ &\quad \frac{N^2}{n^2} \sum_{i=1}^m \frac{y_i^2 (1-p_i)}{p_i} \end{aligned}$$

$$= 2918.429$$

ขั้นที่ 6 คำนวณค่าประมาณยอดรวมประชากร $T_e = \frac{N}{n(1-p_0)} \sum_{i=1}^m \frac{y_i}{\pi_i(m)}$

$$= 314.137$$

$$\begin{aligned} \text{คำนวณ } \hat{V}_3(T_e) &= \frac{N(N-n)}{n(n-1)} \left[\sum_{i=1}^m \frac{y_i^2}{\pi_i(m)} - \frac{1}{n} \left(\sum_{i=1}^m \frac{y_i^2}{\pi_i(m)} + 2 \sum_{i < j} \frac{y_i y_j}{\pi_{ij}(m)} \right) \right] + \\ &\quad \frac{N^2}{n^2(1-p_0)^2} \sum_{i < j} \frac{\pi_i(m)\pi_j(m) - \pi_{ij}(m)}{\pi_{ij}(m)} \left(\frac{y_i - y_j}{\pi_i(m)\pi_j(m)} \right)^2 \end{aligned}$$

$$= 328.871$$

$$\text{คำนวณ } \hat{V}_4(T_e) = N^2 \left(\frac{1}{n} - \frac{1}{N} \right) s^2 + \frac{N^2}{n^2 p^2} (n p q + q) s^2$$

$$\begin{aligned} \text{โดยที่ } s^2 &= \frac{1}{n-1} \left[\sum_{i=1}^m \frac{y_i^2}{\pi_i(m)} - \frac{1}{n} \left(\sum_{i=1}^m \frac{y_i^2}{\pi_i(m)} + 2 \sum_{i < j} \frac{y_i y_j}{\pi_{ij}(m)} \right) \right] \end{aligned}$$

$$= 342.480$$

ขั้นที่ 7 คำนวณ $E = \hat{V}(T_1) / \hat{V}_3(T_e)$

$$8.877$$

โปรแกรมการคำนวณความแปรปรวนของตัวประมาณค่าสอดคล้องรวมประชากร T_1 กับ T_2 เมื่อแต่ละหน่วย
ในประชากรให้ข้อมูลด้วยความน่าจะเป็นไม่เท่ากัน

```

uses crt;

var loop1,loop2,pop,n,m,i,j,k : longint;
    mean,sd : real;
    pm,p,p0,pi_ij,pi_i,v1,v2,v3,v4,v5 : real;
    data : array [1..1000] of real;
    data_n : array [1..1000] of real;
    data_m : array [1..1000] of real;
    check,ch : array [1..1000] of byte;
    pii : array[1..50] of real;
    piij : array[1..50,1..50] of real;
    pp,ppp : array[1..50] of real;
    z : array[1..120] of byte;
    f : text;

function normal : real;
var sum1 : real;
    l : longint;
begin
    sum1 := 0;
    for l := 1 to 12 do
        sum1 := sum1 + random;
        sum1 := sum1 - 6;
        normal := mean + sum1*sd;
    end;
end;

```

```
procedure make_pop;
var i : longint;
begin
  for i := 1 to n do
    data_n[i] := normal;
end;
```

```
procedure sampling_m;
var i,j : longint;
begin
  for i := 1 to n do
    check[i] := 0;
  for i := 1 to m do
    begin
      repeat
        j := random(n)+1;
      until check[j] = 0;
      data_m[i] := data_n[j];
      pp[i] := ppp[j];
      check[j] := 1;
    end;
  j := m+1;
  for i := 1 to n do
    if check[i] = 0 then
      begin
        pp[j] := ppp[i];
        j := j+1;
      end;
  end;
```

```
end;  
end;  
  
procedure make_pi;  
var sum1 : real;  
    l,i : longint;
```

```
begin  
    for i := 1 to n do  
        begin  
            repeat  
                sum1 := 0;  
                for l := 1 to 12 do  
                    sum1 := sum1 + random;  
                sum1 := sum1 - 6;  
            until (sum1 >= 0.5) and (sum1 <= 0.9);  
            ppp[i] := sum1;  
        end;  
    end;  
end;
```

```
procedure find_pm;  
var iii,ii,jj,i,j,k,mm,c : integer;  
    s1,s2,s3,s4 : real;  
    check : boolean;
```

```
begin  
    s4 := 0;
```

```
s1 := pp[1];
for i := 2 to n do
  s1 := s1 * pp[i];
i := 1;
z[i] := 1;
if m <> n then
begin
check := true;
mm := n-m;
repeat
while i <> mm do
begin
  j := i;
  i := i+1;
  z[i] := z[j] + 1;
end;
s2 := s1;
for k := 1 to mm do
begin
  s2 := s2 / pp[z[k]] * (1-pp[z[k]]);
end;
s4 := s4 + s2;
while (z[i]=i+n-mm) and (check = true) do
begin
  i := i-1;
  if i = 0 then
    check := false;
```

```
end;
if check <> false then
z[i] := z[i] + 1;
until check = false;
pm := s4;
end else
  pm := s1;
for iii := 1 to m do
begin
  s4 := 0;
  s3 := pp[n];
  pp[n] := pp[iiii];
  pp[iiii] := s3;
  i := 1;
  z[i] := 1;
  if m <> n then
  begin
    check := true;
    mm := n-m;
    repeat
      while i <> mm do
      begin
        j := i;
        i := i+1;
        z[i] := z[j] + 1;
      end;
    until check = false;
  end;
  s2 := s1;
```



```

for k := 1 to mm do
begin
    s2 := s2 / pp[z[k]] * (1-pp[z[k]]);
end;
s4 := s4 + s2;
while (z[i]=i+n-mm-1) and (check = true) do
begin
    i := i-1;
    if i = 0 then
        check := false;
    end;
    if check <> false then
        z[i] := z[i] + 1;
    until check = false;
    pii[iii] := s4/pm;
end else
    pii[iii] := s1/pm;
    s3 := pp[n];
    pp[n] := pp[iii];
    pp[iii] := s3;
end;
for jj := 1 to m-1 do
begin
    for ii := jj+1 to m do
    begin
        s4 := 0;
        if jj = n-2 then

```

```
begin
  if ii = n-1 then
    begin
      s3 := pp[jj];
      pp[jj] := pp[n];
      pp[n] := s3;
    end else
    begin
      s3 := pp[jj];
      pp[jj] := pp[n-1];
      pp[n-1] := s3;
    end;
  end else
    if jj < n-2 then
      if ii = n-1 then
        begin
          s3 := pp[jj];
          pp[jj] := pp[n];
          pp[n] := s3;
        end else
          if ii = n then
            begin
              s3 := pp[jj];
              pp[jj] := pp[n-1];
              pp[n-1] := s3;
            end else
              begin
```

```
s3 := pp[i1];  
pp[i1] := pp[n];  
pp[n] := s3;  
s3 := pp[jj];  
pp[jj] := pp[n-1];  
pp[n-1] := s3;  
end;  
i := 1;  
z[i] := 1;  
if m <> n then  
begin  
  check := true;  
  mm := n-m;  
  repeat  
    while i <> mm do  
      begin  
        j := i;  
        i := i+1;  
        z[i] := z[j] + 1;  
      end;  
      s2 := s1;  
      for k := 1 to mm do  
        begin  
          s2 := s2 / pp[z[k]] * (1-pp[z[k]]);  
        end;  
        s4 := s4 + s2;  
        while (z[i]=i+n-mm-2) and (check = true) do
```

```

begin
  i := i-1;
  if i = 0 then
    check := false;
  end;
  if check <> false then
    z[i] := z[i] + 1;
  until check = false;
  piij[jj,ii] := s4/pm;
end else
  piij[jj,ii] := s1/pm;
  if jj = n-2 then
  begin
    if ii = n-1 then
    begin
      s3 := pp[jj];
      pp[jj] := pp[n];
      pp[n] := s3;
    end else
    begin
      s3 := pp[jj];
      pp[jj] := pp[n-1];
      pp[n-1] := s3;
    end;
  end else
  if jj < n-2 then
    if ii = n-1 then

```

```
begin
  s3 := pp[jj];
  pp[jj] := pp[n];
  pp[n] := s3;
end else
if ii = n then
begin
  s3 := pp[jj];
  pp[jj] := pp[n-1];
  pp[n-1] := s3;
end else
begin
  s3 := pp[ii];
  pp[ii] := pp[n];
  pp[n] := s3;
  s3 := pp[jj];
  pp[jj] := pp[n-1];
  pp[n-1] := s3;
end;
end;
end;
end;

function v_t_1_1 : real;
var i,j,k : longint;
    s1,s2,s3 : real;
```

```

begin
  s1 := 0;
  for i := 1 to m do
    begin
      s1 := s1 + data_m[i] * data_m[i] * (1-pp[i]) / (pp[i]*pp[i]);
    end;
  v_t_1_1 := s1;
end;

function v_t_2_1 : real;
var i,j,k : integer;
    s1,s2,s3 : real;
begin
  s1 := 0;
  s2 := 0;
  s3 := 0;
  find_pm;
  for i := 1 to m-1 do
    for j := i+1 to m do
      begin
        s3 := data_m[i]/pii[i] - data_m[j]/pii[j];
        s1 := s1 + (pii[i]*pii[j] - piij[i,j])/piij[i,j] * s3 * s3;
      end;
    end;
  s2 := 1-pp[1];
  for i := 2 to m do
    s2 := s2 * (1-pp[i]);
  end;
  s2 := 1-s2;
end;

```

```

s2 := s2*s2;
v_t_2_1 := s1/s2;

```

```

end;

```

```

function v_t_1 : real;

```

```

var i,j,k : longint;

```

```

    s1,s2,s3 : real;

```

```

begin

```

```

    s1 := 0;

```

```

    for i := 1 to m do

```

```

        s1 := s1 + data_m[i]*data_m[i];

```

```

    s3 := s1 * pop * pop *(1-p)/(n * n * p * p);

```

```

    s1 := s1 / p;

```

```

    s2 := 0;

```

```

    for i := 1 to m-1 do

```

```

        for j := i+1 to m do

```

```

            s2 := s2 + data_m[i] * data_m[j];

```

```

    s2 := s2 * 2 / (p*p);

```

```

    s2 := s1 - (s1 + s2)/n;

```

```

    v_t_1 := pop * (pop - n) / (n * (n-1)) * s2 + s3;

```

```

end;

```

```

function v_t_2 : real;

```

```

var i,j,k : longint;

```

```

    s1,s2,s3 : real;

```

```

begin

```

```

    s1 := 0;

```

```

s2 := 0;
s3 := 0;
for i := 1 to m-1 do
  for j := i+1 to m do
    begin
      s1 := s1 + data_m[i] * data_m[j];
      s2 := s2 + (data_m[i]-data_m[j]) * (data_m[i]-data_m[j]);
    end;
  s1 := 2 * s1 / pi_ij;
  s2 := s2*(1/pi_ij - 1/(pi_i*pi_i));
  s2 := s2 * pop * pop / (n * n * (1-p0) * (1-p0));
  for i := 1 to m do
    s3 := s3 + data_m[i] * data_m[i];
  s3 := s3 / pi_i;
  s1 := s3 - (s3 + s1)/n;
  s1 := s1 * pop * (pop-n) / (n*(n-1));
  v_t_2 := s1 + s2;
end;

```

```

function v_t_3 : real;
var i,j,k : longint;
    s1,s2,s3 : real;
begin
  s3 := 0;
  for i := 1 to m-1 do
    for j := i+1 to m do
      s3 := s3 + data_m[i] * data_m[j];
    end;
  end;

```



```

s3 := s3 * 2 / pi_ij;
s1 := 0;
for i := 1 to m do
    s1 := s1 + data_m[i];
s1 := s1 / pi_i ;
s1 := s1 + s3;
s1 := s1 / (n-1);
s2 := 0;
for i := 1 to m do
    s2 := s2 + data_m[i] * data_m[i];
s2 := s2 / pi_i;
s3 := s2 - (s3 + s2)/n;
s2 := pop * (pop - n)/(n * (n-1)) * s3;
s1 := s3/(n-1);
s1 := s2 + pop * pop * (n * p * (1-p) + (1-p))/(n*n*p*p)*s1;
v_t_3 := s1;
end;

procedure read_data;
begin
    repeat
        write('n : ');
        readln(n);
    until n <= 100;
    write('Mean : ');
    readln(mean);
    repeat

```

```
    write('Sd : ');
    readln(sd);
until sd >= 0;
repeat
    write('Loop1 : ');
    readln(loop1);
until loop1 > 0;
repeat
    write('Loop2 : ');
    readln(loop2);
until loop2 > 0;
pop := n;
end;

procedure value;
var i : longint;
begin
    p := m/n;
    pi_i := m/n;
    pi_ij := m*(m-1)/(n*(n-1));
    p0 := 1-p;
    for i := 1 to n-1 do
    begin
        p0 := p0 * (1-p);
    end;
end;
end;
```

```

begin
  clrscr;
  randomize;
  assign(f,'result');
  rewrite(f);
  read_data;
  writeln(f,'n = ',n);
  writeln(f,'Mean = ',mean:7:3);
  writeln(f,'Sd = ',sd:7:3);
  writeln(f);
  writeln('n = ',n);
  writeln('Mean = ',mean:7:3);
  writeln('Sd = ',sd:7:3);
  writeln;
  make_pop;
  make_pi;
  for i := 1 to n do
  begin
    writeln(data_n[i]:10:2,':',ppp[i]:10:2);
  end;
  writeln(f,' m   rep           V(T1)           V(T2)           V(T3)
  writeln(' m   rep           V(T1)           V(T2)           V(T3)
  for i := 1 to 1000 do
    chfil := 0;
  for i := 1 to loop1 do
  begin
    writeln(f);

```

```
writeln;
repeat
  m := random(n div 2)+(n div 2);
until ch[m] = 0;
write(f,m:3);
write(m:3);
value;
ch[m] :=1;
for j := 1 to loop2 do
begin
  sampling_m;
  v1 := v_t_1;
  v2 := v_t_2;
  v3 := v_t_3;
  v4 := v_t_1_1;
  v5 := v_t_2_1;
  if j = 1 then
  begin
    write(f,j:4);
    write(j:4);
  end else
  begin
    write(f,j:7);
    write(j:7);
  end;
  writeln(f,'      ',v1:18:3,v2:18:3,v3:18:3,v1/v2:10:3);
  writeln('      ',v1:18:3,v2:18:3,v3:18:3,v1/v2:10:3);
```

```
writeln(f,'          ',v4:18:3,v5:18:3,v4/v5:28:3);
writeln('          ',v4:18:3,v5:18:3,v4/v5:28:3);

end;

end;

close(f);

end.

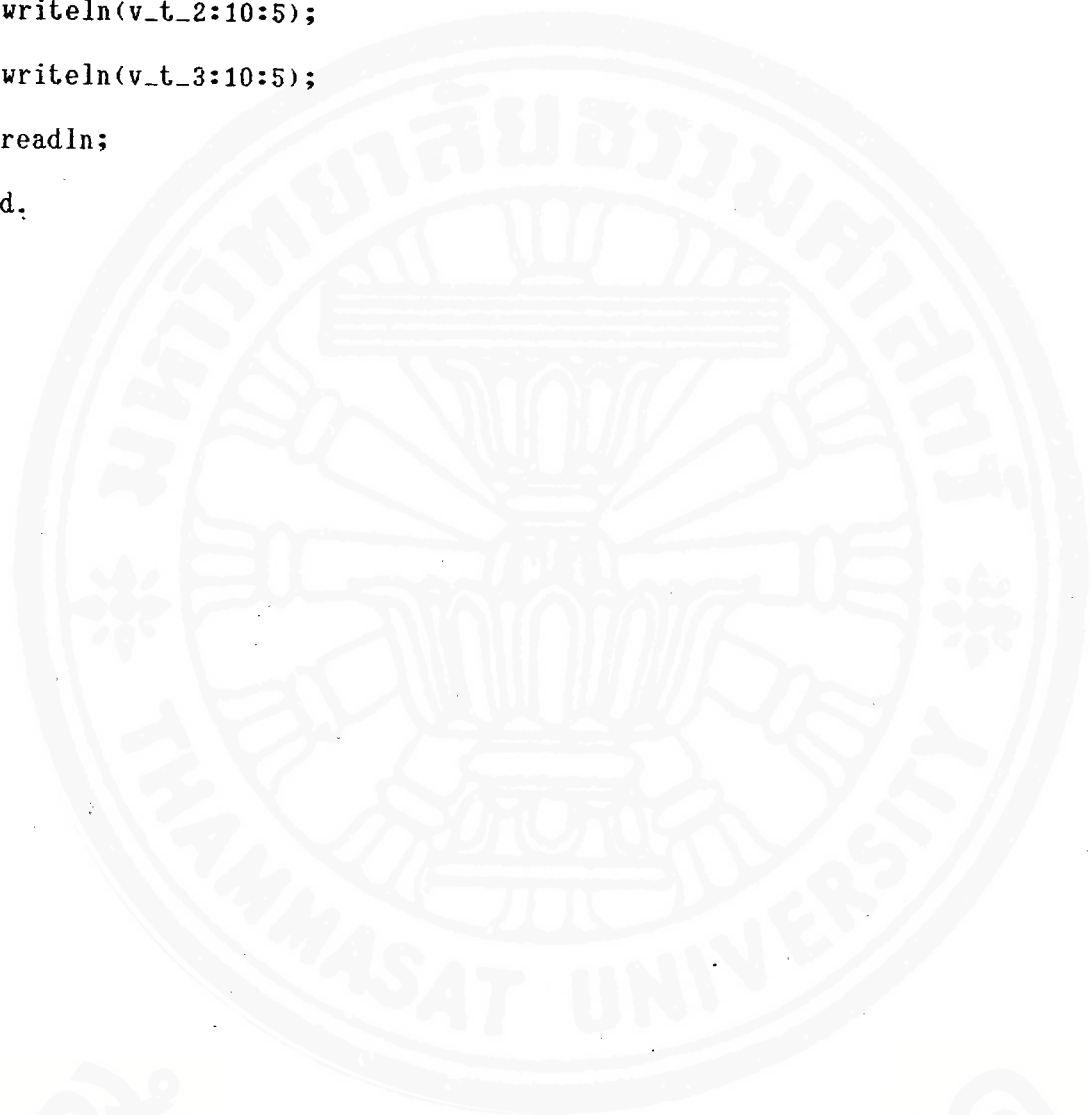
}

begin
  i := 0;
  data_m[1] := 1;
  data_m[2] := 2;
  data_m[3] := 3;
  pp[1] := 0.7;
  pp[2] := 0.7;
  pp[3] := 0.8;
  pp[4] := 0.9;
  pp[5] := 0.2;
  pp[6] := 0.3;
  pp[7] := 0.5;

  m := 2;
  n := 4;
  pop := 4;
  find_pm;

  writeln(v_t_1_1:10:5);
```

```
writeln(v_t_2_1:10:5);  
readln;  
value;  
writeln(v_t_1:10:5);  
writeln(v_t_2:10:5);  
writeln(v_t_3:10:5);  
readln;  
end.
```



สำนักหอสมุด

ตัวอย่าง การที่ใช้โปรแกรมคำนวณความแปรปรวนของตัวประมาณค่ายอดรวมประชากร T_1 กับ T_2 เมื่อแต่ละหน่วยในประชากรให้ข้อมูลด้วยความน่าจะเป็นไม่เท่ากัน

ต้องการศึกษาค่าประมาณยอดรวมประชากร จากตัวอย่างขนาด 10 ซึ่งสุ่มมาโดยไม่ใส่คืน จากประชากรขนาด 6 ที่มีการแจกแจงแบบปกติ ค่าเฉลี่ย 10.000 และความแปรปรวน 9.000 มีรายละเอียดดังนี้

ขั้นที่ 1 สุ่มประชากรขนาด 6 ที่มีการแจกแจงแบบปกติ โดยมีค่าเฉลี่ย 10.00 ความแปรปรวน 9.00 พร้อมทั้งค่าความน่าจะเป็นที่แต่ละหน่วยจะให้ข้อมูล ได้ข้อมูลดังนี้

$$y_i : 10.36 \quad 12.77 \quad 5.21 \quad 8.17 \quad 6.58 \quad 7.72$$

$$p_i : 0.65 \quad 0.81 \quad 0.61 \quad 0.85 \quad 0.50 \quad 0.67$$

ค่ายอดรวมประชากร คือ 50.81

ขั้นที่ 2 จากประชากรในขั้นที่ 1. สมมติว่ามีจำนวนหน่วยที่ให้ข้อมูล (๓) = 4 ได้แก่

$$y_i : 8.17 \quad 10.36 \quad 7.72 \quad 5.21$$

$$p_i : 0.85 \quad 0.65 \quad 0.67 \quad 0.61$$

ก. กรณีที่ทุกหน่วยของประชากรมีความน่าจะเป็นที่จะให้ข้อมูลเท่ากัน

เนื่องจากเป็นการเลือกตัวอย่างแบบ binomial sampling ดังนั้น $p=2/3=0.67$

$$\text{ขั้นที่ 1} \quad \text{คำนวณค่าประมาณยอดรวมประชากร } T_1 = \frac{N}{n} \sum_{i=1}^m y_i p_i$$

$$= 47.187$$

$$\text{คำนวณ } \hat{V}(T_1) = N^2 \left(\frac{1}{n} - \frac{1}{N} \right) s^2 + \frac{N^2}{n^2} \sum_{i=1}^m \frac{y_i^2 (1-p_i)}{p_i}$$

$$\text{เนื่องจาก } N = n = 6 \quad \text{จะได้ว่า } N^2 \left(\frac{1}{n} - \frac{1}{N} \right) s^2 = 0$$

$$\text{ดังนั้น } \hat{V}_0(T_1) = \frac{\sum_{i=1}^m y_i^2 (1-p_i)}{p_i} = 195.621$$

$$\text{ขั้นที่ 2 } \text{ค่าความ } 1-p_0 = 1 - (q_1 q_2 q_3 q_4 q_5 q_6) = 0.009$$

$$(1-p_0)^2 = 0.00008$$

$$T_1(4) = 0.67 = T_i(4), \quad i = 3, 4, 6$$

$$T_{13}(4) = 0.40 = T_{ij}(4), \quad i \neq j = 1, 3, 4, 6$$

$$\text{ขั้นที่ 3 } \text{ค่าความค่าประมาณยอดรวมประชากร } T_2 = \frac{N}{n(1-p_0)} \sum_{i=1}^m \frac{y_i}{T_i(m)}$$

$$= 47.252$$

$$\text{ค่าความ } \hat{V}_{3..}(T_2) = \frac{N(N-n)}{n(n-1)} \left[\frac{\sum_{i=1}^m y_i^2}{\sum_{i=1}^m T_i(m)} - \frac{1}{n} \left(\frac{\sum_{i=1}^m y_i^2}{\sum_{i=1}^m T_i(m)} \right)^2 + 2 \frac{\sum_{i < j}^{mm} y_i y_j}{\sum_{i < j}^{mm} T_{ij}(m)} \right] +$$

$$\frac{N^2}{n^2 (1-p_0)^2} \sum_{i < j}^{mm} \frac{T_i(m) T_j(m) - T_{ij}(m)}{T_{ij}(m)} \left(\frac{y_i}{T_i(m)} - \frac{y_j}{T_j(m)} \right)^2$$

$$13.460$$

$$\text{ค่าความ } \hat{V}_4(T_2) = N^2 \left(\frac{1}{n} - \frac{1}{N} \right) s^2 + \frac{N^2}{n^2 p^2} (npq+q) s^2$$

$$\text{โดยที่ } s^2 = \frac{1}{n-1} \left[\frac{\sum_{i=1}^m y_i^2}{\sum_{i=1}^m T_i(m)} - \frac{1}{n} \left(\frac{\sum_{i=1}^m y_i^2}{\sum_{i=1}^m T_i(m)} \right)^2 + 2 \frac{\sum_{i < j}^{mm} y_i y_j}{\sum_{i < j}^{mm} T_{ij}(m)} \right]$$

$$= 16.779$$

$$E_2 = \hat{V}(T_1) / \hat{V}_{3..}(T_2)$$

$$= 14.534$$

๗. กรณีที่ทุกหน่วยของประชากรมีความน่าจะเป็นที่จะให้ข้อมูลไม่เท่ากัน

$$\text{ขั้นที่ 1 } \text{ค่าความค่าประมาณยอดรวมประชากร } T_1 = \frac{N}{n} \sum_{i=1}^m y_i$$

$$= 45.406$$

$$\text{ค่าความ} \quad \hat{V}(T_1) = N^2 \left(\frac{1}{n} - \frac{1}{N} \right) s^2 + \frac{N^2}{n^2} \sum_{i=1}^m \frac{y_i^2 (1-p_i)}{p_i}$$

เนื่องจาก $N = n = 6$ จะได้ว่า $N^2 \left(\frac{1}{n} - \frac{1}{N} \right) s^2 = 0$

ดังนั้น $\hat{V}_o(T_1) = \frac{\sum_{i=1}^m y_i^2 (1-p_i)}{p_i} = 171.366$

ขั้นที่ 2 ค่าความ $1-p_o = 1 - (q_1 q_2 q_3 q_4 q_5 q_6) = 0.0094$

$$(1-p_o)^2 = 0.00009$$

$$T_1(4) = 0.86$$

$$T_3(4) = 0.58$$

$$T_4(4) = 0.66$$

$$T_6(4) = 0.64$$

ขั้นที่ 3 ค่าความค่าประมาณยอดรวมประชากร $T_2 = \frac{N}{n(1-p_o)} \sum_{i=1}^m \frac{y_i}{T_i(m)}$

$$= 46.425$$

ค่าความ $\hat{V}_{s,u}(T_2) = \frac{N(N-n)}{n(n-1)} \left[\sum_{i=1}^m \frac{y_i^2}{T_i(m)} - \frac{1}{n} \left(\sum_{i=1}^m \frac{y_i^2}{q_i(m)} + 2 \sum_{i < j} \frac{y_i y_j}{T_{i,j}(m)} \right) \right] +$

$$\frac{N^2}{n^2 (1-p_o)^2} \sum_{i < j} \frac{T_i(m) T_j(m) - T_{i,j}(m)}{T_{i,j}(m)} \left(\frac{y_i}{T_i(m)} - \frac{y_j}{T_j(m)} \right)^2$$

$$= 17.581$$

$$E_u = V(T_1) / V_{s,u}(T_2)$$

$$= 9.747$$