VERIFYING DATA HOMOGENEITY. APPLIED PROGRAM *U* TEST (MANN-WITNEY) FOR COMPARING TWO EMPIRICAL DISTRIBUTIONS

Mihai-Radu COSTESCU, Assoc. Prof. PhD.
University of Craiova

Key-words: test, Mann-Witney, procedure.

Abstract: In the practice of investigation, we sometimes encounter the situation of comparing two series of data (samples), or, in other words, appreciating if they have or do not have the same distribution – specified or not –, if they come from the same statistical population which represents them both or not.

We will refer in the following at a test, *U* (*Mann-Witney*), which allows such a comparison, and in the end we will present a using program for this test.

In order to find an answer for the question regarding data homogeneity, there are various methods – statistical tests – more complicated or more efficient.

The parametrical tests, which assume determining some indicators of the two distributions and then comparing them, belong to the first category.

The non parametrical tests, based on rank distribution or on maximal existing distances between the two structures, are – at least apparently – simpler. Their main disadvantage resides in the fact that they are less accurate, because, when using ranks and signs instead of values they tend to lose the quantitative information. Still, accepting this gap, they present the major advantage that the normality of compared distributions is not imposed in their application, thus facilitating its use in comparing ordinal variables and especially the nominal ones.

U test (Mann-Witney) – non parametrical – allows comparing two empirical distributions obtained based on data from two capacity samples n_1 and n_2.

We test thus the hypothesis:

H_0: *there are no significant differences between the two distributions*;

with the alternative

H_1: *the two distributions differ significantly*.

The two data lines corresponding to the two samples form one line. This line is ordered increasingly or decreasingly. Ranks are given to the elements of this line (in the case in which two or more terms have the same rank, these will be assigned the rank average corresponding to them). The rank sum is calculated for each series R_1, R_2, and then two statistics are being calculated:

$$U_1 = n_1 \cdot n_2 + \frac{n_1(n_1+1)}{2} - R_1 \quad (1)$$

$$U_2 = n_1 \cdot n_2 + \frac{n_2(n_2+1)}{2} - R_2\,. \quad (2)$$

Detaining the lowest value *U* from the two previously calculated (if the line has been ordered increasingly), or the highest (if the line has been ordered decreasingly), we calculate the variable:

$$z = \left| \frac{U - \frac{n_1 \cdot n_2}{2}}{\sqrt{\frac{n_1 \cdot n_2 (n_1 + n_2 + 1)}{12}}} \right| \qquad (3)$$

which is compared to the value *zp* corresponding to the chosen probability $(P = 95\%, z = 1{,}96; p = 99\%, z = 2{,}58; P = 99{,}9\%, z = 3{,}3)$.

The critical field is given by $z \geq z_P$, case in which the hypothesis, according to which between the two distributions there are no differences, is rejected, and the trust field of $z < z_P$, case in which the hypothesis that between the two distributions there are no differences is accepted.

The program for applying the test – actually the procedure which can be attached to any program – is the following:

```
procedure mann_witney;
{ Author: Mihai – Radu Costescu
The procedure allows comparing two distributions based on two samples.
Parameter signification:
      n1= capacity of the first sample;
      n2= capacity of the second sample;
      x1= vector of maximum capacity of 100 components
             (values of the first sample);
      x2= vector of maximum capacity of 100 components
             (values of the second sample).
We will work with a probability of 95 %, 99 % or 99,9 %,
the choice rests for the user to decide.}
type vect=array[1..100] of real;
    vect1=array[1..200] of real;
var n1,n2,k:integer;
     x1,x2:vect;
     x,rg:vect1;

procedure citire_siruri(k:integer;var m:integer;var z:vect);
{ reads the dimension of a line and its components }
 var i:integer;
 begin
     write(' introduce the line dimension ',k,'  n=');
     readln(m);
     for i:=1 to m do
        begin
            write('x',k,'(',i,')=');
            readln(z[i]);
        end;
 end;{procedura citire_siruri}

procedure comasare_siruri(n1,n2:integer;x1,x2:vect;var y:vect1);
{ merges the two lines into one }
```

```
 var i:integer;
 begin
     for i:=1 to n1 do
        y[i]:=x1[i];
     for i:=1 to n2 do
        y[n1+i]:=x2[i];
 end;{procedura comasare_siruri}

procedure ordonare_crescatoare(n:integer; var y:vect1);
{ orders increasingly the line obtained from merging the two initial lines
in order to calculate the ranks }
 var
   t,flag:real;
   i:integer;
 begin {procedura ordonare_crescatoare}
     repeat
          flag:=0;
          for i:=1 to n-1 do
             if y[i]>y[i+1] then
                           begin
                                  t:=y[i];
                                 y[i]:=y[i+1];
                                 y[i+1]:=t;
                                  flag:=1;
                           end;
     until flag=0;
 end; {procedura ordonare_crescatoare}

procedure stabilire_ranguri(n:integer;var x,rg:vect1);
{ establishes ranks for the values in the merged line }
 label et1;
 var f,sum:vect;
      h,k,i,l,s:integer;
 begin
     h:=1;
     f[h]:=1;
     for i:=1 to n-1 do
        begin
            if x[i]=x[i+1] then f[h]:=f[h]+1
                          else begin
                                        h:=h+1;
                                        f[h]:=1;
                                end;
        end;
     i:=1;
     k:=1;
     repeat
          s:=0;
          l:=1;
et1:      s:=s+i;
          if l<>f[k] then begin
                                     i:=i+1;
                                    l:=l+1;
                                   goto et1;
                            end
                     else sum[k]:=s;
          k:=k+1;
          i:=i+1;
     until k>h;
     k:=1;
     for i:=1 to n-1 do
       begin
           rg[i]:=sum[k]/f[k];
           if x[i]<>x[i+1] then k:=k+1;
       end;
     if x[n]=x[n-1] then rg[n]:=rg[n-1]
                      else rg[n]:=sum[h]/f[h];
end;{procedura stabilire_ranguri}

procedure determinare_statistica_u(n1,n2:integer;x1,x2:vect;x,rg:vect1);
{ determines U statistics and appreciates homogeneity with a given probability }
 var z,zcalc,op,r1,r2,u,u1,u2:real;
      i,k:integer;
 begin
     writeln(' Choose the probability you want to work with:');
     writeln('   p=95%     give op=1');
     writeln('   p=99%     give op=2');
     writeln('   p=99.9%   give op=3');
     write('                 op = ');
     readln(op);
     if op=1 then z:=1.96
               else if op=2 then z:=2.58
                                else z:=3.3;
     r1:=0;
     for i:= 1 to n1 do
        begin
            k:=0;
            repeat
                 k:=k+1;
            until x1[i]=x[k];
            r1:=r1+rg[k];
        end;
     r2:=0;
     for i:= 1 to n2 do
        begin
            k:=0;
            repeat
                k:=k+1;
            until x2[i]=x[k];
            r2:=r2+rg[k];
        end;
    u1:=n1*n2+n1*(n1+1)/2-r1;
    u2:=n1*n2+n2*(n2+1)/2-r2;
    if u1<u2 then u:=u1
                else u:=u2;
    zcalc:=abs((u-n1*n2/2)/sqrt(n1*n2*(n1+n2+1)/12));
    writeln('z=',zcalc:6:2);
    if zcalc<z then writeln(' There are no differences between the two distributions '
                 else writeln(' There are differences between distributions ');
end;{procedura determinare_statistica_u}

begin {procedure mann_witney}
    k:=1;
    citire_siruri(k,n1,x1);
    k:=2;
    citire_siruri(k,n2,x2);
    comasare_siruri(n1,n2,x1,x2,x);
    ordonare_crescatoare(n1+n2,x);
    stabilire_ranguri(n1+n2,x,rg);
    determinare_statistica_u(n1,n2,x1,x2,x,rg);
end;  {procedure mann_witney}
```

REFERENCES

1. Costescu Mihai-Radu, 2007, *Metode statistice aplicate în ştiinţele sociale*, Casa de Presă şi Editură „Libertatea”, Panciova – Serbia;

2. Costescu Mihai-Radu, Ionescu Adela, 2004, *Prelucrarea informaţională a datelor de măsurare,* Editura Universitaria, Craiova.