stichting
mathematisch
centrum

MC

AFDELING MATHEMATISCHE STATISTIEK SN 10/79 SEPTEMBER
(DEPARTMENT OF MATHEMATICAL STATISTICS)

J.M. BUHRMAN

THE COMPUTATION OF AN UNCONDITIONAL CRITICAL LEVEL WHEN TESTING THE EQUALITY OF TWO UNKNOWN PROBABILITIES

Statal Report 4

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O).

1980 Mathematics subject classification: 62XX04, 62F04

The computation of an unconditional critical level when testing the equality of two unknown probabilities

by

J.M. Buhrman

ABSTRACT

In this report the details are described of the computation of an unconditional critical level in the case of a modified Fisher-Yates-Irwin test for 2×2 contingency tables. This computation is performed by the STATAL program BIN2.

First a description of the principles, due to BOSCHLOO [1], is given. In Section 2 some remarks are made on the concept of unconditional critical levels and the definition in this particular case is given. Section 3 contains a description of the computations and in Section 4 some results are presented. Section 5 consists of the description of some procedures involved and Section 6 gives the ALGOL 60 text of the program.

KEY WORDS & PHRASES: *Bernoulli trials, binomial distributions, small sample tests.*

CONTENTS

1. INTRODUCTION

The STATAL program BIN2 is meant for testing the equality of two probabilities. Let $\underline{x}_1$ and $\underline{x}_2$ have binomial distributions with parameters (n_1,p_1) and (n_2,p_2) respectively. If H: $p_1 = p_2$ is true, the conditional distribution of $\underline{x}_1$ given $\underline{x}_1 + \underline{x}_2 = r$ is hypergeometric, i.e.

$$(1.1) \qquad P(\underline{x}_1 = x \mid \underline{x}_1 + \underline{x}_2 = r) = \binom{n_1}{x}\binom{n_2}{r-x}\binom{n_1+n_2}{r}^{-1}$$

If $\underline{x}_1 + \underline{x}_2 = r$ and $\underline{x}_1$ is in the level α critical region of the hypergeometric distribution with parameters $n_1 + n_2$, n_1 and r, then H is rejected. Boschloo suggested not to use the level α hypergeometric critical regions when a level α test is wanted, but to use level α' hypergeometric critical regions, with $\alpha' > \alpha$. Boschloo's tables indicate what α' must be used in a variety of combinations of n_1, n_2 and α. The value of α' guarantees, that the unconditional test is at level α.

2. THEORETICAL BACKGROUND

Define

$$(2.1a) \qquad p_r(x) = P(\underline{x}_1 \le x \mid \underline{x}_1 + \underline{x}_2 = r)$$ [1)2)]

$$(2.1b) \qquad q_r(x) = P(\underline{x}_1 \ge x \mid \underline{x}_1 + \underline{x}_2 = r)$$

$$(2.1c) \qquad t_r(x) = \min(p_r(x),\ q_r(x))$$

$$(2.2) \qquad \mathrm{bin}(x,n,p) = \binom{n}{x} p^x (1-p)^{n-x}$$

$$(2.3) \qquad Z_1(\varepsilon) = \{(x_1,x_2):\ p_{x_1+x_2}(x_1) \le \varepsilon\}$$

---

1) All probabilities under H.

2) All letters h, i, j, n, r, x with, without or in subscripts are integers.

With this definition $Z_1(\varepsilon)$ is the critical region of the one-sided Fisher test with level ε. Let $h_r(\varepsilon)$ be the left-hand critical value (with level ε) of the hypergeometric distribution with parameters $n = n_1 + n_2$, n_1 and r. Then

$$(2.4) \qquad (x_1,x_2) \in Z_1(\varepsilon) \Longleftrightarrow x_1 \le h_{x_1+x_2}(\varepsilon)$$

However, the difference between ε and the probability of $(\underline{x}_1,\underline{x}_2) \in Z_1(\varepsilon)$ can be considerable, especially for small values of n_1 and n_2, which is due to the discrete character of the hypergeometric distribution. Under $p_1 = p_2 = p$ this probability is

$$(2.5) \qquad P((x_1,x_2) \in Z_1(\varepsilon) \mid p) = \sum_{r=0}^{n} p_r(h_r(\varepsilon))\, \mathrm{bin}(r,n,p) < \varepsilon$$

Notice, that $p_r(h_r(\varepsilon))$ can be equal to ε for some r, but not for all.

BOSCHLOO [1] suggested a method, aiming a level α test, with the use of the same Fisher test, but with a raised (conditional) level. The question is: what level α' should be taken for the Fisher test in order to obtain a level α test, i.e. what α' maximizes the power with the restriction

$$(2.6) \qquad P((\underline{x}_1,\underline{x}_2) \in Z_1(\alpha') \mid p) \le \alpha \qquad \text{for all } p$$

Define

$$(2.7) \qquad \alpha(\alpha') = \max_p P((\underline{x}_1,\underline{x}_2) \in Z_1(\alpha') \mid p)$$

Notice, that α is a step function, which is constant on intervals $[\alpha_1',\alpha_2')$ with $Z_1(\alpha')$ invariant for $\alpha' \in [\alpha_1',\alpha_2')$.

The critical level of an outcome is defined as the smallest level at which the hypothesis can be rejected with this outcome (see e.g. [2]). Now, notice, that the smallest α' for which $(x_1,x_2) \in Z_1(\alpha')$, is given by $p_{x_1+x_2}(x_1)$ in the one-sided case. So the critical level of the outcome (x_1,x_2) is

$$(2.8) \qquad \alpha(p_{x_1+x_2}(x_1)) = \max_p P((\underline{x}_1,\underline{x}_2) \in Z_1(p_{x_1+x_2}(x_1)) \mid p)$$

The critical level in the one-sided case will be denoted by $\gamma_1(x_1,x_2)$.

For the two-sided case we define

$$(2.9) \qquad Z_2(\varepsilon) = \{(x_1,x_2)\colon t_{x_1+x_2}(x_1) \le \tfrac{1}{2}\varepsilon\}$$

Further, let $k_r(\varepsilon)$ be the right-hand critical value (with level ε) of the hypergeometric distribution with parameters $n = n_1+n_2$, n_1 and r. Then, writing $r = x_1+x_2$ for now and for the sequel,

$$(2.10) \qquad (x_1,x_2) \in Z_2(\varepsilon) \Longleftrightarrow x_1 \le h_r(\tfrac{1}{2}\varepsilon) \vee x_1 \ge k_r(\tfrac{1}{2}\varepsilon)$$

The two-sided critical level, denoted by $\gamma_2(x_1,x_2)$, then follows

$$(2.11) \qquad \gamma_2(x_1,x_2) = \max_p P((\underline{x}_1,\underline{x}_2) \in Z_2(2t_r(x_1)) \mid p)$$

Further, it should be noticed, that, if $x_1/n_1 < x_2/n_2$ (i.e. $x_1 < E(\underline{x}_1 \mid \underline{x}_1+\underline{x}_2 = r)$ and x_1 is in the left-hand tail of the hypergeometric distribution), then γ_2 is in general smaller than $2\gamma_1$, except if $n_1 = n_2$, which can be seen by considering (write $\alpha' = p_r(x_1)$)

$$(2.12a) \qquad P((\underline{x}_1,\underline{x}_2) \in Z_2(2\alpha') \mid p) =$$

$$= \sum_{j=0}^{n} P(\underline{x}_1 \le h_j(\alpha') \vee \underline{x}_1 \ge k_j(\alpha') \mid \underline{x}_1+\underline{x}_2 = j)\ \mathrm{bin}(j,n,p) =$$

$$(2.12b) \qquad = \sum_{j=0}^{n} p_j(h_j(\alpha'))\ \mathrm{bin}(j,n,p) +$$

$$(2.12c) \qquad + \sum_{j=0}^{n} q_j(k_j(\alpha'))\ \mathrm{bin}(j,n,p)$$

If (2.12b) reaches its maximum for p_1, then, by symmetry, (2.12c) has the same maximum in $1-p_1$. In general (2.12a) has its maximum neither in p_1 nor in $1-p_1$. Then $\gamma_2(x_1,x_2) = \max(2.12a) < \max(2.12b) + \max(2.12c) = 2\gamma_1(x_1,x_2)$. However, if $n_1 = n_2$, then either $p_1 = 1-p_1 = \frac{1}{2}$ and (2.12a) has its maximum in $p = \frac{1}{2}$ as well or (2.12b) has a maximum for $p_1 \ne \frac{1}{2}$, but then, again by symmetry, it reaches the same maximum in $1-p_1$, and (2.12c) does the same.

3. COMPUTATIONAL SCHEME

The program begins with the input of n_1, x_1, n_2, x_2 and i, where x_1 and x_2 are the observed values of $\underline{x}_1$ and $\underline{x}_2$ respectively and i is 1 (or 2) if a one-sided (or two-sided) test is required. The sum $x_1 + x_2$ is denoted by r, and $n_1 + n_2$ is denoted by n.

3.1. Transformation of the observations (lines 165-168)[1]

In the two-sided case the check is made if $x_1/n_1 \leq x_2/n_2$. If not, (x_1,n_1) and (x_2,n_2) are interchanged. So we have either $x_1/n_1 \leq x_2/n_2$ or the one-sided case, for which the alternative is $p_1 < p_2$.

3.2. Computation of the conditional critical level (lines 172-176)

If $n > 50$, the value of $\alpha' = p_r(x_1)$ is computed by *hyperg*(x_1,n_1,r,n)[2]. If $n \leq 50$, the value of α' is computed as a fraction, of which the numerator and denominator are computed from the binomial coefficients (see (1.1)). The reason for this very accurate computation will be revealed in subsection 3.5.

3.3. Quick treatment of special cases (lines 178-183)

If $x_1 = 0$ and $x_2 = n_2$ in the one-sided case, $p_0 = x_2/n$ maximizes (2.8) and $\gamma_1 = \alpha'\ \text{bin}(n_2,n,p_0)$.

If $x_1 = 0$ and $x_2 = n_2$ and $n_1 = n_2$ in the two-sided case, $p_0 = \frac{1}{2}$ maximizes (2.11) and $\gamma_2 = 2\alpha'\ \text{bin}(n_2,n,\frac{1}{2})$. (See also Section 5 sub *range*.)

If $x_1/n_1 = x_2/n_2$ in the two-sided case, then $\gamma_2 = 1$, which is obvious.

If $x_2 = 0$ or $x_1 = n_1$, then $\gamma_1 = 1$.

---

[1]) Line numbers refer to the program, Section 6.

[2]) Procedures are described in Section 5. Procedure names are written in italics.

3.4. Approximation for $n > 200$ (lines 184-185)

If $n > 200$

$$\gamma_i = i \times \{\frac{2}{3}\,hyperg(x_1,r,n_1,n) + \frac{1}{3}\,hyperg(x_1-1,r,n_1,n)\}$$

which is based on Boschloo's rule of thumb (see [1]).

3.5. The remaining cases (lines 186-206)

The conditional critical values $h_j(\alpha')$ are determined and stored in array cv[0:n] for $j = 1,2,...,n-1$ and the corresponding probabilities $p_j(h_j(\alpha'))$ are stored in array pr[0:n] (lines 188-189). By definition the conditional critical regions for $j = 0$ and $j = n$ are empty and pr[0] = pr[n] = 0 (line 187). The determination of cv[1: n-1] is done by the procedure "*critical values small*" (lines 103-121) for $n \le 50$, and by "*critical values large*" (lines 93-102) for $50 < n \le 200$. In the latter case cv[j] is determined by the procedure *hypinv* as the maximum k that satisfies

$$(3.1) \qquad hyperg(k,n_1,j,n) \le \alpha' + 10^{-12}$$

The term 10^{-12} is necessary to prevent errors due to rounding errors. It turned out, that in case of omitting 10^{-12}, for some j the possibility occurred that cv[j] became k-1, where $hyperg(k,n_1,r,n)$ was exactly equal to α'. This is of course not a very elegant way of coping with machine errors. Therefore the probabilities involved are computed as the ratio of two integers (lines 80-92) if possible, i.e. if $n \le 50$. Instead of comparing two probabilities as in (3.1), the numerators of two probabilities with equal denominators are compared.

In the one-sided case $pr[j] = p_j(h_j(\alpha'))$ is used directly in the following function of p

$$(3.2) \qquad s(p) = \sum_{j=1}^{n-1} pr[j]\, bin(j,n,p)$$

which is computed in the procedure "*size*" (lines 139-148). In the two-sided case (lines 130-138), by symmetry,

$$pr[j] = p_j(h_j(\alpha')) + q_j(k_j(\alpha')) = p_j(h_j(\alpha')) + p_{n-j}(h_{n-j}(\alpha'))$$

Now it remains to maximize s(p), see (2.11) and (2.12). Let V be a finite subset[1)] of the interval (0,1) and p_0' maximizes s(p) on V. Then

$$s(p_0') \leq \gamma_i = \max\{s(p): p \in [0,1]\} < i\alpha'$$

These inequalities provide the initial values, m_1 and M_1, of belowmax (lines 197-198) and abovemax (lines 191-192) respectively. Usually m_1 is much closer to γ_i than M_1. Therefore a second try is $g = 0.98m_1 + 0.02M_1$. If the function s(p) - g has no zeros, then max s(p) < g and the process is repeated with $m_2 = m_1$ and $M_2 = g$, otherwise max s(p) ≥ g and the process is repeated with $m_2 = g$ and $M_2 = M_1$. The process stops as soon as $M_j - m_j \leq 10^{-4} m_j$ (lines 199-204). Then the critical level γ_i gets the value $\frac{1}{2}m_j + \frac{1}{2}M_j$, m_j and M_j being the final under- and overestimate respectively. The search for zeros is done by the procedure *zeroin*.

4. ACCURACY AND COMPUTATION TIME

The stopping rule described at the end of Section 3 guarantees a relative precision of 10^{-4}, i.e. the relative error is less than 10^{-4}. However, owing to other inaccuracies it can occur that the relative error is a bit more but probably less than 10^{-3}. For practical purposes this is sufficiently accurate. In the case n > 200 an approximating procedure is used, which gives a much less accurate result. An impression of its merits is obtained by cases where the exact and approximating procedure have both been applied (see Table 4.1).

The computation time depends strongly on n_1 and n_2, particularly due to the procedure "*size*", which requires about n steps. For this reason an approximate value is calculated for n > 200. Table 4.1 shows several results. Both conditional and unconditional critical levels are given to

---

1) In the program V has 21 elements.

illustrate the differences. The computation was done on a CDC, Cyber 73-28/ 173-12 (SARA, Amsterdam). The computation times are given in the "seconds" used in this system. They have no absolute value. Identical computations need different times presumably due to the time-sharing features of the system.

Input					Output			
n_1	x_1	n_2	x_2	i	α'	γ_i	appr.	comp. time
5	2	5	3	1	.5	.376954		.73
5	1	5	2	1	.5	.376963		.99
5	0	5	5	1	.003968	.000977		.01
15	5	5	2	1	.594169	.409552		1.00
25	12	5	2	1	.567050	.405706		1.38
35	10	5	2	1	.477240	.319266		1.66
55	15	5	2	1	.439612	.295427		3.76
75	25	5	2	1	.551749	.401697		5.01
95	35	5	2	1	.613901	.470888		6.19
60	11	40	12	1	.132604	.096495	.107	5.33
100	16	20	5	1	.250986	.179860	.201	6.28
100	16	50	13	1	.108125	.081369	.088	7.95
10	5	8	2	1	.278281	.164664		.86
10	5	8	2	2	.278281	.307233		.89
10	2	10	3	1	.5	.411918		.98/.65*
10	2	10	3	2	.5	.823800		.97/.65*
25	10	25	18	1	.022502	.013220		2.31
25	10	25	18	2	.022502	.026412		2.98

* The computation times for identical cases differ remarkably.

Table 4.1. Results of BIN2: critical levels, approximations and computation times.

5. PROCEDURES USED

Procedures from NUMAL [3] and STATAL [4] are described briefly.

zeroin, code number 34436 (NUMAL)

The boolean procedure *zeroin*(x,y,fx,tol) determines whether the function fx (expression containing x) has zeros in the interval between the values of x and y at the moment *zeroin* is called for. tol is a precision parameter.

hyperg, code number 41004 (STATAL)

The value of *hyperg*(x,n_1,r,n) is

$$\sum_{j=0}^{x} \binom{n}{j}\binom{n-n_1}{r-j}\binom{n}{r}^{-1}$$

hypinv, code number 41005 (STATAL)

The value of *hypinv*(a,n_1,r,n,"true") is

$$\max\{k: hyperg(k,n_1,r,n) \le a\}$$

binpro, code number 41251 (STATAL)

The value of *binpro*(x,n,p) is

$$\binom{n}{x}p^x(1-p)^{n-x}$$

Out of the procedures defined in the program some are explained below, the other ones are believed not to need any more explanation.

hypergnumer (lines 80-92)

The value of *hypergnumer*(x,r) is

$$\sum_{j=0}^{x} \binom{r}{j}\binom{n-r}{n_1-j}$$

bincoef (lines 149-156)

The value of *bincoef*(n,j) is $\binom{n}{j}$

range (lines 122-129)

The procedure *range* determines the interval $[r_1,r_2]$ of values of j for which $p_j(h_j(\alpha'))$ is positive. By analytical reasoning follows that s(p) attains its maximum for $r_1/n \le p \le r_2/n$ in the one-sided case and for $\min(r_1,n-r_2)/n \le p \le \max(r_2,n-r_1)$ in the two-sided case.

6. PROGRAM TEXT

```
"BEGIN" "COMMENT" THIS PROGRAM TESTS THE HYPOTHESIS OF THE EQUALITY
    OF TWO PROBABILITIES. THE RESULTING CRITICAL LEVEL IS OBTAINED
    BY BOSCHLOO'S MODIFICATION OF THE FISHER-IRWIN TEST.
*-------------------------------------------------------------------*;
"INTEGER" N1, X1, N2, X2, R1, R2, N, ALPHANUMER, ALPHADENOM, R, RS,
    ENDINPUT, ERRORCOUNTER;
"INTEGER" "ARRAY" TXT[1:10];
"BOOLEAN" ONE SIDED, CORNER, DIAGONAL, ERR, APPROX;
"REAL" P, P1, P2, ST, ALPHA, OK, BELOWMAX, ABOVEMAX, GOK, SIZE1,
      X, Y, X1H, X2H, N1H, N2H, SIDES;
"BOOLEAN" "PROCEDURE" ZEROIN(PAR); "CODE" 34436;
"REAL" "PROCEDURE" HYPERG(PAR); "CODE" 41004;
"REAL" "PROCEDURE" HYPINV(PAR); "CODE" 41005;
"REAL" "PROCEDURE" BINPRO(PAR); "CODE" 41251;
"PROCEDURE" MCVIGNET(CHN, TXT); "CODE" 40203;
"REAL" "PROCEDURE" MAX2(A,B); "VALUE" A,B; "REAL" A,B;
    MAX2 := "IF" A > B "THEN" A "ELSE" B;
"REAL" "PROCEDURE" MIN2(A,B); "VALUE" A,B; "REAL" A,B;
    MIN2 := "IF" A < B "THEN" A "ELSE" B;
"COMMENT" SPECIFIC PROCEDURES ARE DECLARED BELOW
*-------------------------------------------------------------------*;
"PROCEDURE" OBSERVATIONS;
"BEGIN"
    ERR := "FALSE";
    INPUT(60,"("10(8A)")", TXT);
    "IF" TXT[1] = ENDINPUT "THEN" "GO TO" EINDE "ELSE"
        OUTPUT(61,"(" 10(8A),2/ ")", TXT);
    INPUT(60,"(" 2(N) ")", N1H, X1H);
    "IF" N1H < X1H "OR" X1H < 0 "OR" ENTIER(X1H) < X1H "OR"
         ENTIER(N1H) < N1H "THEN"
    "BEGIN"
        OUTPUT(61,"(" "("THE PROGRAM EXPECTS THE NUMBER OF ")",
        "("EXPERIMENTS OF THE FIRST TYPE AND ")", /,
        "("THE CORRESPONDING NUMBER OF SUCCESSES. HOWEVER, ")",
        "("YOU GAVE")", -7ZD.DD, /, "("AND")", -7ZD.DD,
        "(" RESPECTIVELY.")",/")", N1H, X1H);
        ERR := "TRUE";
    "END" "ELSE"
    "BEGIN" X1 := X1H; N1 := N1H;
        OUTPUT(61,"("/,"("EXPERIMENT TYPE 1:")",7ZDB,")", X1);
        "IF" X1=1 "THEN" OUTPUT(61,"(""("SUCCESS   IN")"")")
                  "ELSE" OUTPUT(61,"(""("SUCCESSES IN")"")");
        OUTPUT(61,"("7ZDB,"("EXPERIMENT")"")", N1);
        "IF" N1=1 "THEN" OUTPUT(61,"(""(".")",2/")")
                  "ELSE" OUTPUT(61,"(""("S.")",2/")");
    "END";
    INPUT(60,"(" 2(N) ")", N2H, X2H);
    "IF" N2H < X2H "OR" X2H < 0 "OR" ENTIER(X2H) < X2H "OR"
         ENTIER(N2H) < N2H "THEN"
    "BEGIN"
        OUTPUT(61,"(" "("THE PROGRAM EXPECTS THE NUMBER OF ")",
        "("EXPERIMENTS OF THE SECOND TYPE AND ")", /,
        "("THE CORRESPONDING NUMBER OF SUCCESSES. HOWEVER, ")",
        "("YOU GAVE")", -7ZD.DD, /, "("AND")", -7ZD.DD,
        "(" RESPECTIVELY.")", / ")", N2H, X2H);
        ERR := "TRUE";
    "END" "ELSE"
```

```
    "BEGIN" X2 := X2H; N2 := N2H;
        OUTPUT(61,"("/,"("EXPERIMENT TYPE 2:")",7ZDB,")", X2);
        "IF" X2=1 "THEN" OUTPUT(61,"(""("SUCCESS   IN")"")")
                  "ELSE" OUTPUT(61,"(""("SUCCESSES IN")"")");
        OUTPUT(61,"("7ZDB,"("EXPERIMENT")"")", N2);
        "IF" N2=1 "THEN" OUTPUT(61,"(""(".")",2/")")
                  "ELSE" OUTPUT(61,"(""("S.")",2/")");
    "END";
    INPUT(60, "("N,/")", SIDES);
    "IF" SIDES ^= 1 "AND" SIDES ^= 2 "THEN"
    "BEGIN" OUTPUT(61, "(""("THE PROGRAM EXPECTS THE DIGIT")",
        "(" 1 (FOR A ONE-SIDED TEST)")",/,"("OR 2 (FOR A TW")",
        "("O-SIDED TEST)")", 6/ ")");
        ERR := "TRUE";
    "END" "ELSE"
    "IF" ERR "THEN" OUTPUT(61, "(" 7/ ")") "ELSE"
        ONE SIDED := SIDES=1;
    "IF" ERR "THEN"
        "BEGIN" ERRORCOUNTER := ERRORCOUNTER + 1 ; "GO TO" ERROR "END"
        "ELSE" ERRORCOUNTER := 0;
"END" OBSERVATIONS
*-----------------------------------------------------------------*;
"INTEGER" "PROCEDURE" HYPERGNUMER(X,R);
    "VALUE" X,R; "INTEGER" X,R;
"IF" X<0 "OR" X<R+N1-N "THEN" HYPERGNUMER := 0 "ELSE"
"BEGIN" "INTEGER" J, SUM, TERM;
    SUM := TERM := BINCOEF(N-R, N1-X) * BINCOEF(R,X);
    J := X+1;
    "FOR" J := J-1 "WHILE" TERM > 0 "DO"
    "BEGIN" TERM := TERM * J * (N-R-N1+J) / (R-J+1) / (N1-J+1);
        SUM := SUM + TERM;
    "END";
    HYPERGNUMER := SUM;
"END" HYPERGNUMER
*-----------------------------------------------------------------*;
"PROCEDURE" CRITICAL VALUES LARGE(CV, PR, PROB);
    "VALUE" PROB; "REAL" PROB; "ARRAY" CV, PR;
"BEGIN" "INTEGER" R;
    PROB := PROB + "-12;
    "FOR" R := 1 "STEP" 1 "UNTIL" N-1 "DO"
    "BEGIN" CV[R] := HYPINV(PROB, N1, R, N, "TRUE");
        PR[R] := HYPERG(CV[R], N1, R, N);
    "END";
"END" CRITICAL VALUES LARGE
*-----------------------------------------------------------------*;
```

```
"PROCEDURE" CRITICAL VALUES SMALL(CV, PR, TAIL);
    "VALUE" TAIL; "INTEGER" TAIL; "ARRAY" CV, PR;
"BEGIN" "INTEGER" R, K, H, TEL; "REAL" PROB;
    PROB := TAIL / ALPHADENOM;
    "FOR" R := 1 "STEP" 1 "UNTIL" N-1 "DO"
    "BEGIN" H := HYPINV(PROB, N1, R, N, "TRUE");
        TEL := HYPERGNUMER(H,R);
        "FOR" K := H "WHILE" TEL <= TAIL "DO"
        "BEGIN" H := H+1;
            TEL := TEL + BINCOEF(R,H) * BINCOEF(N-R,N1-H);
        "END";
        "FOR" K := H "WHILE" TEL > TAIL "DO"
        "BEGIN" TEL := TEL - BINCOEF(R,H) * BINCOEF(N-R,N1-H);
            H := H-1;
        "END";
        CV[R] := H; PR[R] := TEL / ALPHADENOM;
    "END";
"END" CRITICAL VALUES SMALL
*----------------------------------------------------------------------*;
"PROCEDURE" RANGE(CV, PR); "ARRAY" CV, PR;
"BEGIN" "INTEGER" R;
    R := R1 := R2 := N2;
    "FOR" R := R-1 "WHILE" CV[R] >= 0 "DO" R1 := R;
    R := N2;
    "FOR" R := R+1 "WHILE" CV[R] >= R+N1-N "DO" R2 := R;
"END" RANGE
*----------------------------------------------------------------------*;
"PROCEDURE" TWO SIDED(PR); "ARRAY" PR;
"BEGIN" "INTEGER" R, K;
    R1 := MIN2(R1,N-R2);
    R2 := N-R1;
    K := ENTIER((N+.5)/2);
    "FOR" R := R1 "STEP" 1 "UNTIL" K "DO"
    PR[N-R] := PR[R] := PR[R] + PR[N-R];
"END" TWO SIDED
*----------------------------------------------------------------------*;
"REAL" "PROCEDURE" SIZE(PR,P); "VALUE" P; "REAL" P; "ARRAY" PR;
"BEGIN" "INTEGER" R; "REAL" BINPR,S,Q;
    S := 0; Q := 1-P; BINPR := BINPRO(R1,N,P);
    "FOR" R := R1 "STEP" 1 "UNTIL" R2 "DO"
    "BEGIN" S := S + BINPR*PR[R];
          BINPR := BINPR * P / Q * (N-R) / (R+1);
    "END";
    SIZE := S;
"END" SIZE
*----------------------------------------------------------------------*;
"INTEGER" "PROCEDURE" BINCOEF(N,K); "VALUE" N,K; "INTEGER" N,K;
"BEGIN" "INTEGER" B,L,K1;
    K1 := "IF" K > N-K "THEN" N-K "ELSE" K;
    B := 1;
    "FOR" L := 1 "STEP" 1 "UNTIL" K1 "DO" B := B*(N+1-L)//L;
    BINCOEF := B;
"END" BINCOEF
*----------------------------------------------------------------------*;
NOW THE PROGRAM BEGINS
:
MCVIGNET(61,"("BIN2 : TEST OF EQUALITY OF TWO PROBABILITIES")");
OUTPUT(61,"("*")");
ENDINPUT := EQUIV("("ENDINPUT")");
ERRORCOUNTER := 0;
```

```
NEXT PROBLEM:
OBSERVATIONS;
"IF" X1/N1 > X2/N2 "AND" "NOT" ONE SIDED "THEN"
"BEGIN" R := X1; X1 := X2; X2 := R;
        R := N1; N1 := N2; N2 := R;
"END";
CORNER := X1=0 "AND" X2=N2;
DIAGONAL := X1/N1 = X2/N2;
N := N1+N2; RS := X1+X2;
"IF" N > 50 "THEN" ALPHA := HYPERG(X1,N1,RS,N) "ELSE"
"BEGIN" ALPHANUMER := HYPERGNUMER(X1,RS);
        ALPHADENOM := BINCOEF(N,N1);
        ALPHA := ALPHANUMER / ALPHADENOM;
"END";
APPROX := N>200;
"IF" CORNER "AND" ONE SIDED "THEN"
OK := ALPHA * BINPRO(N2,N,N2/N) "ELSE"
"IF" CORNER "AND" "NOT" ONE SIDED "AND" N1=N2 "THEN"
OK := ALPHA * BINPRO(N2,N,N2/N) * 2 "ELSE"
"IF" DIAGONAL "AND" "NOT" ONE SIDED "THEN" OK := 1 "ELSE"
"IF" X2 * (N1-X1) = 0 "THEN" OK := 1 "ELSE"
"IF" APPROX "THEN"
    OK := ((2*ALPHA + HYPERG(X1-1,RS,N1,N))/3) * SIDES "ELSE"
"BEGIN" "INTEGER" "ARRAY" CV[0:N]; "REAL" "ARRAY" PR[0:N];
    CV[0] := -1; CV[N] := N1-1; PR[0] := PR[N] := 0;
    "IF" N > 50 "THEN" CRITICAL VALUES LARGE(CV,PR,ALPHA)
        "ELSE" CRITICAL VALUES SMALL(CV,PR,ALPHANUMER);
    RANGE(CV, PR);
    "IF" ONE SIDED "THEN" ABOVEMAX := ALPHA
        "ELSE" "BEGIN" ABOVEMAX := 2 * ALPHA; TWO SIDED(PR) "END";
    BELOWMAX := SIZE1 := SIZE(PR,RS/N);
    P1 := R1/N;
    "IF" ONE SIDED "THEN" P2 := R2/N "ELSE" P2 := .51;
    ST := (P2-P1)/20;
    "FOR" P := P1 "STEP" ST "UNTIL" P2 "DO"
        BELOWMAX := MAX2(SIZE(PR,P), BELOWMAX);
    "FOR" GOK := (ABOVEMAX + BELOWMAX*50) / 51
    "WHILE" ABOVEMAX - BELOWMAX > "-4 * BELOWMAX "DO"
    "BEGIN" X := P1; Y := P2;
        "IF" ZEROIN(X,Y, GOK-SIZE(PR,X), .1/N) "THEN"
        BELOWMAX := MAX2(SIZE(PR,X),SIZE(PR,Y)) "ELSE" ABOVEMAX := GOK;
    "END";
    OK := (ABOVEMAX + BELOWMAX) / 2;
"END";
"IF" APPROX "THEN" OUTPUT(61,"("/,"("APPROX. CRITICAL LEVEL:")",
    3B")") "ELSE" OUTPUT(61,"("/,"("CRITICAL LEVEL:")", 11B")");
OUTPUT(61,"("-ZD.6D, "("  (")",D,"("-SIDED)")",
            //,"("CONDITIONAL CRIT. LEVEL:  ")",-ZD.6D,
               "("  (")",D,"("-SIDED)")",4/")",
          OK, SIDES, MIN2(ALPHA*SIDES , 1) ,  SIDES);
ERROR: "IF" ERRORCOUNTER = 3 "THEN" "GO TO" EINDE;
"GO TO" NEXT PROBLEM;
EINDE:
"END"
```

REFERENCES

[1] BOSCHLOO, R.D. (1970), *Raised conditional level of significance for the* 2×2-*table when testing the equality of two probabilities*, Statistica Neerlandica 24.1, p. 1-9.

[2] LEHMANN, E.L. (1959), *Testing Statistical Hypotheses.*

[3] NUMAL, *A library of numerical procedures in ALGOL 60*, Mathematisch Centrum, Amsterdam.

[4] STATAL, *A library of statistical procedures and programs*, Mathematisch Centrum, Amsterdam.

Index of symbols