Algorithm AS 89: The Upper Tail Probabilities of Spearman's Rho

Author(s): D. J. Best and D. E. Roberts

Source: *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 1975, Vol. 24, No. 3 (1975), pp. 377-379

Published by: Wiley for the Royal Statistical Society

Stable URL: https://www.jstor.org/stable/2347111

Algorithm AS 89

The Upper Tail Probabilities of Spearman's *Rho*

By D. J. Best and D. E. Roberts

C.S.I.R.O. Division of Mathematics and Statistics, North Ryde, Australia

Keywords: SPEARMAN'S RHO; EDGEWORTH SERIES APPROXIMATION; RANK CORRELATION

Language

ISO Fortran

Description and Purpose

Suppose two rankings (without ties), $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_n$, of n objects are given and the value

$$S = \sum_{i=1}^{n} (x_i - y_i)^2$$

calculated. The function computes the probability on the null hypothesis of obtaining a value greater than or equal to S. The statistic, S, is related to *rho* (ρ) by

$$\rho = 1 - \frac{6S}{n^3 - n}$$

and its use in significance tests is conventional.

Numerical Method

For $n < 7$ the probability is calculated exactly by evaluating S for all $n!$ permutations (Langdon, 1967) and forming the cumulative distribution.

For $n \geqslant 7$ a power series expansion in n^{-1} based on the Edgeworth series approximation (David *et al.*, 1951) is used. Table 1 compares this approximation for all possible S for the given n, with (i) the Beta approximation (Pitman, 1937) and (ii) the Pearson curve method (Olds, 1938; Zar, 1972). The Edgeworth series is clearly superior. Algorithm AS 66 (Hill, 1973) was used to evaluate the normal integral.

Table 1

Comparison of approximations: (a) *Edgeworth series*; (b) *Pearson curve method*; (c) *Beta (or t−) approximation*

	n							
	7		9		11		13	
Method	σ	$\lvert\Delta_m\rvert$	σ	$\lvert\Delta_m\rvert$	σ	$\lvert\Delta_m\rvert$	σ	$\lvert\Delta_m\rvert$
(a)	·0025	·0046	·0006	·0011	·0003	·0006	·0002	·0005
(b)	·0025	·0073	·0009	·0025	·0005	·0013	·0004	·0010
(c)	·0043	·0112	·0026	·0059	·0019	·0041	·0015	·0033

σ = standard deviation of errors. Δ_m = maximum error.

Note that the Beta approximation is commonly used to test ρ outside the range of existing tables by referring $t = \rho(n-2)^{\frac{1}{2}}(1-\rho^2)^{-\frac{1}{2}}$ to tables of Student's t with $n-2$ degrees of freedom. An equivalent test which follows from the relation $t = 0{\cdot}5(n-2)^{\frac{1}{2}}(F^{\frac{1}{2}}-F^{-\frac{1}{2}})$ (Cacoullos, 1965) is to refer $F = (1+\rho)(1-\rho)^{-1}$ to tables of Snedecor's F with $n-2$ and $n-2$ degrees of freedom.

Structure

FUNCTION PRHO(N, IS, IFAULT)

Formal parameters

N	Integer	input:	the number of objects ranked
IS	Integer	input:	the value of S
IFAULT	Integer	output:	a fault indicator, equal to: 1 if $n \leqslant 1$, and 0 otherwise

Time and Accuracy

Use of the Edgeworth approximation for $n \geqslant 7$ ensures at least two decimal place accuracy which should be adequate for many purposes.

For $n \geqslant 7$ one call to *PRHO* requires about $1{\cdot}5 \times 10^{-4}$ seconds of execution time while $n < 7$ requires at most 9×10^{-3} seconds on a CDC 7600.

The user may wish to calculate exact values for $n > 6$ and of course this is easily done by changing the *DIMENSION* statement and the statement

IF (*N.GT.*6) *GOTO* 6

appropriately. It should be stressed, however, that the time to calculate an exact probability increases dramatically. For example, for $n = 8$ the execution time for one exact calculation is 0·57 seconds on a CDC 7600. Based on figures supplied by a referee an IBM 370/158 (using the optimizing compiler) would take about 3·1 seconds to do the same calculation while an ICL 4-75 using the fastest compiler would take about 21·0 seconds. Similar comments concerning execution time also apply to other methods which have been proposed for generating the exact distribution.

Acknowledgements

The authors wish to thank a referee for demonstrating the consequences of using the algorithm to obtain exact probabilities for $n \geqslant 7$ and also to thank the Algorithm Editor for his suggestions.

References

Cacoullos, T. (1965). A relation between the t and F-distributions. *J. Amer. Statist. Ass.*, **60**, 528–531.

David, S. T., Kendall, M. G. and Stuart, A. (1951). Some questions of distribution in the theory of rank correlation. *Biometrika*, **38**, 131–140.

Hill, I. D. (1973). Algorithm AS 66: The normal integral. *Appl. Statist.*, **22**, 424–427.

Langdon, G. G. (1967). An algorithm for generating permutations. *Commun. Ass. Comput. Mach.*, **10**, 298–299.

Olds, E. G. (1938). Distributions of sums of squares of rank differences for small numbers of individuals. *Ann. Math. Statist.*, **9**, 133–148.

Pitman, E. J. G. (1937). Significance tests which may be applied to any samples from any populations. II. The correlation coefficient test. *J. R. Statist. Soc. Supplement*, **4**, 225–232.

Zar, J. H. (1972). Significance testing of the Spearman rank correlation coefficient. *J. Amer. Statist. Ass.*, **67**, 578–580.

```
      FUNCTION PRHO(N, IS, IFAULT)
C
C        ALGORITHM AS 89  APPL. STATIST. (1975) VOL.24, NO.3
C
C        TO EVALUATE THE PROBABILITY OF OBTAINING A VALUE GREATER THAN OR
C        EQUAL TO IS, WHERE IS=(N**3-N)*(1-R)/6, R=SPEARMANS RHO AND N
C        MUST BE GREATER THAN 1
C
      DIMENSION L(6)
C
C        TEST ADMISSIBILITY OF ARGUMENTS AND INITIALIZE
C
      PRHO = 1.0
      IFAULT = 1
      IF (N .LE. 1) RETURN
      IFAULT = 0
      IF (IS .LE. 0) RETURN
      PRHO = 0.0
      IF (IS .GT. N * (N * N - 1) / 3) RETURN
      JS = IS
      IF (JS .NE. 2 * (JS / 2)) JS = JS + 1
      IF (N .GT. 6) GOTO 6
C
C        EXACT EVALUATION OF PROBABILITY
C
      NFAC = 1
      DO 1 I = 1, N
      NFAC = NFAC * I
      L(I) = I
    1 CONTINUE
      PRHO = 1.0 / FLOAT(NFAC)
      IF (JS .EQ. N * (N * N - 1) / 3) RETURN
      IFR = 0
      DO 5 M = 1, NFAC
      ISE = 0
      DO 2 I = 1, N
      ISE = ISE + (I - L(I)) ** 2
    2 CONTINUE
      IF (JS .LE. ISE) IFR = IFR + 1
      N1 = N
    3 MT = L(1)
      NN = N1 - 1
      DO 4 I = 1, NN
      L(I) = L(I + 1)
    4 CONTINUE
      L(N1) = MT
      IF (L(N1) .NE. N1 .OR. N1 .EQ. 2) GOTO 5
      N1 = N1 - 1
      IF (M .NE. NFAC) GOTO 3
    5 CONTINUE
      PRHO = FLOAT(IFR) / FLOAT(NFAC)
      RETURN
C
C        EVALUATION BY EDGEWORTH SERIES APPROXIMATION
C
    6 B = 1.0 / FLOAT(N)
      X = (6.0 * (FLOAT(JS) - 1.0) * B / (1.0 / (B * B) - 1.0) - 1.0)
     *    * SQRT(1.0 / B - 1.0)
      Y = X * X
      U = X * B * (0.2274 + B * (0.2531 + 0.1745 * B) + Y * (-0.0758
     *   + B * (0.1033 + 0.3932 * B) - Y * B * (0.0879 + 0.0151 * B
     *   - Y * (0.0072 - 0.0831 * B + Y * B * (0.0131 - 0.00046 * Y)))))
C
C        CALL TO ALGORITHM AS 66
C
      PRHO = U / EXP(Y / 2.0) + ALNORM(X, .TRUE.)
      IF (PRHO .LT. 0.0) PRHO = 0.0
      IF (PRHO .GT. 1.0) PRHO = 1.0
      RETURN
      END
```