Algorithm AS 157

The Runs-Up and Runs-Down Tests

By R. G. T. GRAFTON

Department of Computational and Statistical Science, Liverpool University, UK†

Keywords: RUNS-UP AND RUNS-DOWN; STATISTICAL TEST OF RANDOMNESS; PSEUDO-RANDOM; RANDOM NUMBER GENERATOR

LANGUAGE

Fortran 66

DESCRIPTION AND PURPOSE

Given a sequence of length n, the runs-up and runs-down tests can be applied simultaneously as an effective test of randomness (Knuth, 1969). For a precise definition of a run consider the following example. Taking a sequence of 12 *distinct* numbers 2, 7, 8, 1, 9, 6, 4, 0, 3, 11, 10, 17; and starting at the beginning of the sequence, let a run-up break-point be any point at which $X_{j+1} < X_j$. The number of observations (r) between successive break-points is defined as a run-up of length r. In the above sequence we mark each such break-point by a hyphen, obtaining 2, 7, 8-1, 9-6-4-0, 3, 11-10, 17 giving runs-up of lengths 3, 2, 1, 1, 3 and 2. In a similar way, for runs-down let a break-point be any point at which $X_{j+1} > X_j$. The example has runs-down of lengths 1, 1, 2, 4, 1, 2 and 1.

The subroutine $UDRUNS$ simultaneously calculates the number of runs-up and runs-down of length i for $i = 1(1)5$ and the number of runs-up and runs-down of length greater than or equal to 6. For example, the number of runs-up of length i for $i = 1(1)5$ is stored in $UCOUNT(1), \ldots, UCOUNT(5)$, and the number of runs-up of length greater than or equal to 6 is stored in $UCOUNT(6)$. The following statistic is then computed:

$$UV = n^{-1} \sum_{i=1}^{6} \sum_{j=1}^{6} (UCOUNT(i) - nb_i)(UCOUNT(j) - nb_j)\, a_{i,j},$$

where the coefficients $a_{i,j}$ and b_i are:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,6} \\ a_{2,1} & & & \vdots \\ & & & \vdots \\ & & & \vdots \\ & & & \vdots \\ a_{6,1} & \cdots & \cdots & a_{6,6} \end{bmatrix} = \begin{bmatrix} 4529{\cdot}4 & 9044{\cdot}9 & 13568 & 18091 & 22615 & 27892 \\ & 18097 & 27139 & 36187 & 45234 & 55789 \\ & & 40721 & 54281 & 67852 & 83685 \\ & & & 72414 & 90470 & 111580 \\ & & & & 113262 & 139476 \\ & & & & & 172860 \end{bmatrix},$$

symmetrical

† Now at Overseas Containers Limited, Beagle House, Braham Street, London E1 8EP, England.

$(b_1, \ldots, b_6) = (\frac{1}{6}, \frac{5}{24}, \frac{11}{120}, \frac{19}{720}, \frac{29}{5040}, \frac{1}{840})$.

The statistic UV has a χ^2 distribution with 6 degrees of freedom when n is large ($n \geqslant 4000$ is the recommendation of Knuth, 1969). Similar calculations are performed for runs-down, resulting in the calculation of the test statistic DV.

An algorithm for a test on the total number of runs in a sequence of n distinct randomly ordered numbers was first published by Downham (1970), but Wedderburn (1976) pointed out that an incorrect test statistic had been calculated. Knuth (1969) shows how to calculate the correct test statistics for the case when n is large and runs-up and runs-down are considered separately. The present algorithm is a Fortran coding of the runs-up and runs-down tests given by Knuth.

Structure

SUBROUTINE UDRUNS (X, N, UV, DV, IFAULT)

Formal parameters

X	Real array (N)	input: the sequence whose randomness is being tested
N	Integer	input: size of the array X
UV	Real	output: the runs-up test statistic
DV	Real	output: the runs-down test statistic
$IFAULT$	Integer	output: if $N < 4000$, $IFAULT = N$
		: adjacent data ties, $IFAULT = 1$
		: otherwise, $IFAULT = 0$

Failure indications

If the routine is used for a sample of size $N < 4000$, the routine will exit with a failure message of $IFAULT = N$. The runs-up and runs-down tests assume that the input sequence consists of distinct numbers, and a simple check for ties between adjacent elements is provided. If this test proves to be positive, then the routine will exit with a failure message of $IFAULT = 1$.

The test statistics for the numbers of runs-up and runs-down in the sample have approximate χ^2 distributions each with 6 degrees of freedom when the sample size is large. Knuth (1969) suggests 4000 as a minimum sample size for these tests. It should be noted that the statistics for the runs-up and runs-down tests are not independent.

Use of Runs-Up and Runs-Down Tests

An important use of the runs-up and runs-down tests is in testing the randomness of a uniformly distributed pseudo-random sequence in (0, 1). Such sequences frequently come from a multiplicative congruential psuedo-random number generator, i.e., a generator of the form

$$x_{i+1} = kx_i \pmod{m}, \quad i = 0, 1, 2, \ldots,$$

where x_0, k, m are integers. The sequence $\{x_i/m\}$ is a uniform pseudo-random sequence in (0, 1). Downham and Roberts (1967) used a number of methods for testing the randomness of uniform sequences generated by a selection of pseudo-random number generators. Their results showed that "runs-up and runs-down" tests were extremely useful and sensitive; although they used the test incorrectly, it is generally thought that their conclusions are still valid (see Lewis *et al.*, 1969).

Golder (1976a) maintains that testing the sequences themselves can prove costly in terms of computer time and storage and that savings can be made if, in the first place, the spectral test is used to screen the pseudo-random number generators. Golder (1976a, b) has misinterpreted Knuth's presentation of the spectral test, and Hoaglin and King (1978) have clarified the circumstances under which the spectral test should be used with multiplicative congruential generators, i.e. generators of the form

(i) m is a prime and k is a primitive root modulo m.
(ii) $m = 2^l$ for some $l \geqslant 3$ and $k \equiv 5 \pmod 8$)

(for case (ii)) the correct spectral test will be produced by using $m/4$ as the input value of the modulus to *SPEC* in Algorithm AS 98.

Use of the spectral test is an effective prelude to sequence testing because not only will fewer sequences need to be tested for randomness, but also poor generators will be instantly excluded. Therefore, it is recommended that the spectral test and the runs-up and runs-down tests be used "in harness".

Test Results

Six multiplicative pseudo-random number generators as quoted by Downham and Roberts (1967) were used to test sequences of length $n = 10\,000$ in order to give examples of the effectiveness of using the spectral test and the runs-up and runs-down tests in harness. (For each generator, m is a prime and k is a primitive root modulo m.)

In the spectral test the quantities C_j (as specified by Golder, 1976a) are calculated for various values of j. For $j = 2, 3, \ldots$, C_j tells how nearly the entire generated sequence approaches the j-dimensional behaviour of a truly random uniform sequence. Golder's routine deals with values of $j = 2, 3, 4$ and 5. Essentially, the larger the value of C_j the more preferable the generator and if C_2, C_3, C_4 and C_5 are all $\geqslant 0{\cdot}1$ then the generator is said to have passed the spectral test (for a full discussion, see Golder, 1976a).

The generators, C_j ($j = 2, 3, 4$ and 5), and the runs-up and runs-down test statistics are given in Table 1.

TABLE 1

Results of the spectral test and runs-up and runs-down tests

	Multiplier	Modulus	Spectral test				Runs test†	
	k	m	C_2	C_3	C_4	C_5	UV	DV
(1)	8 192	67 101 323	3·14	0·0638	3·43	0·2194	11·752*	7·510
(2)	8 192	67 099 547	3·14	0·0360	3·53	0·5744	2·742	3·482
(3)	32 768	16 775 723	0·4665	3·58	1·88	0·4378	6·554	4·127
(4)	54 751	99 707	2·79	2·78	0·0012	0·0030	544·9***	549·3***
(5)	8	67 100 963	0·000003	0·000033	0·00031	0·0027	166·9***	133.7***
(6)	32	7 999 787	0·0004	0·017	0·6481	2·46	13·924**	13·321**

(i) $x_0 = 1001$ in all cases for the runs-up and runs-down tests.
(ii) */**/*** denotes significance at the 10 per cent/5 per cent/1 per cent level, respectively.
† Note that UV and DV are not independent.

The results of Downham and Roberts, the spectral test, and the runs-up and runs-down tests have all shown that generators (4), (5) and (6) are unacceptable. The results of Downham and Roberts show that the first three generators appear quite satisfactory. However, for generators (1) and (2), the spectral test and the runs-up and runs-down tests have produced a slightly different set of conclusions. The spectral test C_3 value is slightly below the minimum acceptance level for both generators and this indicates that, although both may be acceptable for a specific use, further testing is suggested. The runs-up and runs-down tests highlight this fact by rejecting generator (1) and accepting generator (2). Generator (3) has passed both the spectral test and the runs-up and runs-down tests and is considered suitable for general use.

Time

The routine was run on the Manchester University regional computer, a CDC 7600 front-ended by an ICL 1906A. Only approximate timings are available, but total run time consists of two parts, the first part being constant and the second part proportional to the sample size n.

$$\begin{aligned}\text{total run time} &= \text{compilation time} + \text{execution time}\\ &\approx 0{\cdot}285 + 5{\cdot}0 \times 10^{-6} * n \text{ secs}\end{aligned}$$

Thus, for a sample size of 10 000, total run time $\approx 0{\cdot}305$ secs.

Acknowledgements

The author would like to thank the referee for many useful comments and Mr J. G. Settle, Computational and Statistical Science Department, Liverpool University for his advice and help in the presentation of the algorithm. Thanks are also due to the Algorithm Editor for some helpful comments. The author was supported by an S.R.C. Research Grant while carrying out this work.

References

Downham, D. Y. (1970). Algorithm AS 29. The runs up and down test. *Appl. Statist.*, **19**, 190–192.

Downham, D. Y. and Roberts, F. D. K. (1967). Multiplicative congruential pseudo-random number generators. *Computer J.*, **10**, 74–77.

Golder, E. R. (1976a). Algorithm AS 98. The spectral test for the evaluation of congruential pseudo-random generators. *Appl. Statist.*, **25**, 173–180.

—— (1976b). Remark AS R18. The spectral test for the evaluation of congruential pseudo-random generators. *Appl. Statist.*, **25**, 324.

Hoaglin, D. C. and King, M. L. (1978). Remark AS R24. A remark on Algorithm AS 98. The spectral test for the evaluation of congruential pseudo-random generators. *Appl. Statist.*, **27**, 375.

Knuth, D. E. (1969). *The Art of Computer Programming*, Vol. 2, Reading, Mass.: Addison-Wesley.

Lewis, P. A. W., Goodman, A. S. and Miller, J. M. (1969). A pseudo-random number generator for the IBM 360. *IBM Systems J.*, **8**, 136.

Wedderburn, R. W. M. (1976). Remark AS R16. A remark on Algorithm AS 29. The runs up and down test. *Appl. Statist.*, **25**, 193.


```
      SUBROUTINE UDRUNS(X, N, UV, DV, IFAULT)
C
C        ALGORITHM AS 157 APPL. STATIST. (1981) VOL.30, NO.1
C
C        THE RUNS-UP AND RUNS-DOWN TEST
C
      INTEGER UCOUNT, DCOUNT, RU, RD
      DIMENSION X(N), A(6, 6), B(6), UCOUNT(6), DCOUNT(6)
C
C        SET-UP THE A AND B MATRICES
C
      DATA
     *  A(1, 1), A(1, 2), A(1, 3), A(1, 4), A(1, 5), A(1, 6), A(2, 2),
     *  A(2, 3), A(2, 4), A(2, 5), A(2, 6), A(3, 3), A(3, 4), A(3, 5),
     *  A(3, 6), A(4, 4), A(4, 5), A(4, 6), A(5, 5), A(5, 6), A(6, 6)
     *  /4529.4, 9044.9, 13568.0, 18091.0, 22615.0, 27892.0, 18097.0,
     *  27139.0, 36187.0, 45234.0, 55789.0, 40721.0, 54281.0, 67852.0,
     *  83685.0, 72414.0, 90470.0, 111580.0, 113262.0, 139476.0,
     *   172860.0/
      IFAULT = 0
      IF (N .LT. 4000) GOTO 500
      DO 1 J = 2, 6
      J1 = J - 1
      DO 1 I = 1, J1
      A(J, I) = A(I, J)
    1 CONTINUE
      B(1) = 1.0 / 6.0
      B(2) = 5.0 / 24.0
      B(3) = 11.0 / 120.0
      B(4) = 19.0 / 720.0
      B(5) = 29.0 / 5040.0
      B(6) = 1.0 / 840.0
C
      DO 100 I = 1, 6
      UCOUNT(I) = 0
      DCOUNT(I) = 0
  100 CONTINUE
C
C        THE LOOP THAT ENDS AT LINE 300 DETERMINES THE NUMBER OF
C        RUNS-UP AND RUNS-DOWN OF LENGTH I FOR I=1(1)5 AND THE NUMBER
C        OF RUNS-UP AND RUNS-DOWN OF LENGTH GREATER OR EQUAL TO 6
```

```
C
      RU = 1
      RD = 1
      DO 300 J = 2, N
C
C        THE FOLLOWING STATEMENT TESTS FOR BOTH RUNS-UP AND
C        RUNS-DOWN BREAK-POINTS. IF A RUN-DOWN BREAK-POINT IS
C        DETECTED - GOTO 200, OTHERWISE - GOTO 150.
C        ( RU AND RD ACT AS LOCAL COUNTERS FOR THE NUMBER OF
C        RUNS-UP AND RUNS-DOWN RESPECTIVELY.)
C        A TEST IS ALSO MADE FOR DATA TIES BETWEEN ADJACENT
C        ELEMENTS. IF A TIE IS DETECTED - GOTO 600 .
C
      IF (X(J) - X(J - 1)) 150, 600, 200
  150 UCOUNT(RU) = UCOUNT(RU) + 1
      RU = 1
      IF (RD .LT. 6) RD = RD + 1
      GOTO 300
  200 DCOUNT(RD) = DCOUNT(RD) + 1
      RD = 1
      IF (RU .LT. 6) RU = RU + 1
  300 CONTINUE
      UCOUNT(RU) = UCOUNT(RU) + 1
      DCOUNT(RD) = DCOUNT(RD) + 1
C
C        CALCULATE THE TEST STATISTICS UV AND DV.
C
      UV = 0.0
      DV = 0.0
      RN = FLOAT(N)
      DO 400 I = 1, 6
      DO 400 J = 1, 6
      UV = UV + (FLOAT(UCOUNT(I)) - RN * B(I)) *
     *   (FLOAT(UCOUNT(J)) - RN * B(J)) * A(I, J)
      DV = DV + (FLOAT(DCOUNT(I)) - RN * B(I)) *
     *   (FLOAT(DCOUNT(J)) - RN * B(J)) * A(I, J)
  400 CONTINUE
      UV = UV / RN
      DV = DV / RN
      GOTO 700
  500 IFAULT = N
      GOTO 700
  600 IFAULT = 1
  700 RETURN
      END
```

Algorithm AS 158

Calculation of the Probabilities $\{P(l, k)\}$ for the Simply Ordered Alternative

By G. W. CRAN

The Queen's University of Belfast, Northern Ireland

Keywords: MULTIPLE COMPARISON; INFERENCE UNDER ORDER RESTRICTIONS

LANGUAGE

Fortran 66

PURPOSE

Given a random sample of size n_i from $N(\mu_i, a_i\sigma^2)$, $i = 1, \ldots, k, k \geqslant 3$, where the μ's and σ^2 are unknown and the a's are known, a test of

$$H_0\colon \mu_1 = \mu_2 = \ldots = \mu_k$$

against $H_1\colon \mu_1 \leqslant \mu_2 \ldots \leqslant \mu_k$, not all the μ's equal, may be based on Bartholomew's $\bar{E}_k^2$ statistic—see Barlow *et al.* (1972, p. 121). The probability distribution function of $\bar{E}_k^2$ assuming H_0 may be expressed as the weighted sum of incomplete beta function ratios, the weights being the probabilities $\{P(l, k), l = 1, \ldots, k\}$ (Barlow *et al.*, 1972, p. 127). The purpose of the algorithm is the computation of the probabilities $\{P(l, k)\}$, whence $P(\bar{E}_k^2 \geqslant C;\, H_0)$ may be calculated using Algorithm AS 123 (Bremner, 1978).