```
      S23 = F1(W2, W3, W4)
      S34 = F1(W3, W4, W5)
      S45 = F1(W4, W5, W6)
      PR2 = 0.03125 + 0.0625 * PII * (S12 + S23 + S34 + S45) +
     *  0.125 * PII * PII * (S12 * S34 + S12 * S45 + S23 * S45)
      IF (PR2 .LT. 0.0) PR2 = 0.0
      RETURN
      END
C
      FUNCTION F2(I, J, V1, V2, V3, V4, V5)
C
C        ALGORITHM AS 158.4 APPL. STATIST. (1981) VOL.30, NO.1
C
      DIMENSION VV(10)
      VV(1) = V1
      VV(2) = V2
      VV(3) = V3
      VV(4) = V4
      VV(5) = V5
      F2 = PR1(I, J, VV)
      RETURN
      END
C
      FUNCTION FACT(M, IFAULT)
C
C        ALGORITHM AS 158.5 APPL. STATIST. (1981) VOL.30, NO.1
C
C        CALCULATION  OF  M  FACTORIAL
C
      DATA MAXM /50/
      IFAULT = 3
      IF (M .LT. 0 .OR. M .GT. MAXM) RETURN
      IFAULT = 0
      FACT = 1.0
      IF (M .LE. 1) RETURN
      A1 = 1.0
      DO 1 I = 2, M
      AI = I
      A1 = A1 * AI
    1 CONTINUE
      FACT = A1
      RETURN
      END
```

Algorithm AS 159

An Efficient Method of Generating Random $R \times C$ Tables with Given Row and Column Totals

By W. M. PATEFIELD

University of Salford, UK

Keywords: TWO-WAY CONTINGENCY TABLES

LANGUAGE

Fortran 66

DESCRIPTION AND PURPOSE

Purpose

Boyett (AS 144, 1979) describes a subroutine *RCONT* which generates a random table from the exact distribution over all tables with given row and column totals. The subroutine was used by Agresti *et al.* (1979) to approximate significance levels of exact tests on $r \times c$ tables. Their procedure is to sample tables by a large number of calls of *RCONT* and to estimate the significance level by the proportion of samples with a value of a particular test statistic that provides at least as much evidence against the null hypothesis as the value of the statistic for the observed table. Such a procedure is of value when there is doubt about the validity of the asymptotic distribution of the test statistic or when the asymptotic distribution is unknown. In the latter case particularly, this procedure would be used even when N (the total number of observations in the table) is large.

The c.p.u. times for calls to *RCONT* were found by Boyett to be approximately proportional to N. The subroutine *RCONT*2 given here generates a random table by a different method and is typically considerably faster than *RCONT* for larger values of N.

The subroutine *RCONT*2 has the further advantage that, with minor modification, it calculates the probability of each generated table.

Theory

Under the null hypothesis of no association between row and column categories, the joint probability distribution of a table $\{a_{ij},\ 1 \leqslant i \leqslant r,\ 1 \leqslant j \leqslant c\}$ conditional on the row and column totals $(a_{i.},\ 1 \leqslant i \leqslant r;\ a_{.j},\ 1 \leqslant j \leqslant c)$ is quoted by Boyett (1979). From this, the conditional probability distribution of entry a_{lm} given the entries in the previous rows, i.e. $(a_{ij},\ i = 1, \ldots, l-1,\ j = 1, \ldots, c)$ and the previous entries in row l, i.e. $(a_{lj},\ j = 1, \ldots, m-1)$ is found to be

$$\begin{aligned} P_{lm} = {} & \left(a_{l.} - \sum_{j=1}^{m-1} a_{lj}\right)! \left(N - \sum_{i=1}^{l} a_{i.} - \sum_{j=1}^{m-1} a_{.j} + \sum_{j=1}^{m-1} \sum_{i=1}^{l} a_{ij}\right)! \\ & \times \left(a_{.m} - \sum_{i=1}^{l-1} a_{im}\right)! \left\{ \sum_{j=m+1}^{c} \left(a_{.j} - \sum_{i=1}^{l-1} a_{ij}\right)\right\}! \\ & \times \left[a_{lm}! \left(a_{.m} - \sum_{i=1}^{l} a_{im}\right)! \left(a_{l.} - \sum_{j=1}^{m} a_{lj}\right)! \right. \\ & \left. \times \left(N - \sum_{i=1}^{l} a_{i.} - \sum_{j=1}^{m} a_{.j} + \sum_{j=1}^{m} \sum_{i=1}^{l} a_{ij}\right)! \left\{ \sum_{j=m}^{c} \left(a_{.j} - \sum_{i=1}^{l-1} a_{ij}\right)\right\}! \right]^{-1}, \end{aligned} \qquad (1)$$

where $N = \Sigma_{i=1}^{r} a_{i.}$ and $(1 \leqslant l \leqslant r-1, 1 \leqslant m \leqslant c-1)$.

The above conditional probability is valid when either $l = 1$ or $m = 1$ if the convention

$$\sum_{i=1}^{0} (\cdot) = \sum_{j=1}^{0} (\cdot) = 0$$

is employed.

The range of a_{lm} is such that all the factorial terms are non-negative. The conditional expected value of a_{lm} given previous entries and the row and column totals is

$$E_{lm} = \left(a_{.m} - \sum_{i=1}^{l-1} a_{im}\right)\left(a_{l.} - \sum_{j=1}^{m-1} a_{lj}\right) \bigg/ \sum_{j=m}^{c} \left(a_{.j} - \sum_{i=1}^{l-1} a_{ij}\right).$$

When the denominator in this expression is zero, $E_{lm} = 0$.

Numerical method

Although formula (1) for the conditional probability distribution of a_{lm} appears rather complicated, each of the terms is evaluated sequentially as $l = 1, \ldots, r-1$; $m = 1, \ldots, c-1$ in only a few lines of code. For each (l, m) the algorithm generates a random number, *RAND*, between 0 and 1.

The probability distribution of a_{lm} is then accumulated, starting with a_{lm} equal to the nearest integer to E_{lm}. Further elements of this cumulative distribution are computed from this initial element by simple arithmetic. The value of a_{lm} is chosen with the required conditional probability when the cumulative probability exceeds *RAND*. By this procedure (1) is only evaluated once for each (l, m) The advantage of accumulating the distribution about E_{lm} is two fold. Firstly, accuracy is ensured as the extreme elements of the distribution often have near zero conditional probabilities. Secondly, the algorithm only evaluates the part of the cônditional distribution required to accumulate up to *RAND*. These advantages are particularly noticeable when the range of a_{lm} is large.

Finally, the entries a_{rm} $(m = 1, ..., c)$ in the last row and a_{lc} $(l = 1, ..., r)$ in the last column are obtained from the final values of terms in (1).

Structure

SUBROUTINE RCONT2 (NROW, NCOL, NROWT, NCOLT, JWORK, MATRIX, KEY, IFAULT)

Formal parameters

NROW	Integer	input:	number of rows in observed matrix
NCOL	Integer	input:	number of columns in observed matrix
NROWT	Integer array (*NROW*)	input:	vector of row totals of the observed matrix
NCOLT	Integer array (*NCOL*)	input:	vector of column totals of the observed matrix
JWORK	Integer array (*NCOL*)	workspace	
MATRIX	Integer array (*NROW*, *NCOL*)	output:	the randomly generated matrix
KEY	Logical	input:	*KEY* = .*FALSE*. for initial call;
		output:	*RCONT*2 will set *KEY* = .*TRUE*. for subsequent calls. *KEY* must not be changed between successive calls to *RCONT*2.
IFAULT	Integer	output:	fault indicator, equal to: 1 if $NROW \leqslant 1$; 2 if $NCOL \leqslant 1$; 3 if a row total $\leqslant 0$; 4 if a column total $\leqslant 0$; 5 if sample size exceeds 5000; 0 otherwise

Auxiliary function

The Numerical Algorithms Group (N.A.G.) function *GO5AAF*(*DUMMY*) returns a pseudo-random number uniformly distributed between 0 and 1. The variable *DUMMY* is a dummy argument required by the Fortran syntax. This function may be replaced by the user's own random number generator or by the N.A.G. Mark 7 function *GO5CAF*. It is usually called $(r-1)(c-1)$ times for each call to *RCONT*2.

Restrictions and Time

Restrictions

MATRIX must have at least two rows and two columns and all row and column totals must be positive. On the first call, subroutine *RCONT*2 stores $\text{Log}_e I!$ in element $I+1$ of the real array *FACT* for $I = 0, ..., N$. The subroutine is coded with *FACT* of dimension 5001 to allow for a maximum of 5000 observations. To alter the allowable sample size one need only change the dimension of *FACT* and the value given to *MAXTOT* in the data statement. The labelled common "/*B*/*NTOTAL*, *NROWM*, *NCOLM*, *FACT*" is used to maintain the values of the sample size *NTOTAL*, *NROWM* ($= r-1$), *NCOLM* ($= c-1$) and the array of log-factorials between subsequent calls to *RCONT2*.

Time

The following Tables 1 and 2 of times in seconds of c.p.u. per 1000 calls are intended to compare the efficiency of *RCONT* (Boyett, 1979) with that of *RCONT2*. The times were observed using an ICL 1904S which is more than ten times slower than the AMDAHL 470/V6 used by Boyett.

TABLE 1
Seconds of c.p.u. time per 1000 calls to RCONT

Table dimension	*Sample size, N*							
	10	20	30	50	100	200	500	1000
Row × Column								
2 × 7	4·86	9·11	13·20	21·56	42·21	83·63	209·75	414·76
3 × 4	4·53	8·32	11·99	19·45	38·13	75·30	182·41	372·48
4 × 4	4·68	8·47	12·14	19·59	38·28	75·47	185·92	372·82
5 × 5	5·17	9·04	12·91	20·65	39·97	78·60	192·29	387·20
6 × 6	5·54	9·55	13.67	21·58	41·62	81·62	204·76	401·50

TABLE 2
Seconds of c.p.u. time per 1000 calls to RCONT2

Table dimension	*Sample size, N*							
	10	20	30	50	100	200	500	1000
Row × Column								
2 × 7	6·48	6·98	7·23	7·59	8·35	9·27	11·50	13·96
3 × 4	6·45	6·93	7·21	7·58	8·40	9·50	11·87	14·51
4 × 4	9·08	9·89	10·26	10·81	11·83	13·51	16·30	20·02
5 × 5	15·51	16·63	17·15	18·07	19·65	21·87	26·54	31·66
6 × 6	21·31	24·73	25·82	27·01	29·13	32·17	38·42	45·22

As can be seen from the tables, whereas the time required to generate random tables using *RCONT* is approximately proportional to sample size, the time using *RCONT2* is much less dependent on sample size but more dependent on the dimension of the table. For the first two tables with six degrees of freedom, *RCONT2* is more efficient when $N = 20$ or more. For the last table with 25 degrees of freedom *RCONT2* is more efficient only when $N = 100$ or more. These timings were made on tables with all row totals approximately equal to N/r and all column totals approximately equal to N/c. When these totals are unequal the time taken by *RCONT* is almost unchanged, whereas the time taken by *RCONT2* is reduced. Further computations also show that *RCONT2* is at least twice as fast as *RCONT* for the tables studied by Agresti *et al.* (1979).

CALCULATIONS OF THE PROBABILITY OF GENERATED TABLES

From (1), under the null hypothesis of no association between row and column categories, the probability of observing the sampled table is

$$P = \prod_{l=1}^{r-1} \prod_{m=1}^{c-1} P_{lm}$$

$$= \prod_{i=1}^{r} a_{i.}! \prod_{j=1}^{c} a_{.j}! \bigg/ \left\{ N! \prod_{i=1}^{r} \prod_{j=1}^{c} a_{ij}! \right\}. \tag{2}$$

This is the probability given by Boyett (1979). It may be readily computed by *RCONT2* as the product of the conditional probabilities. The resultant probability is returned in the real variable *HOP* of the labelled common "*/TEMPRY/HOP*" if the subroutine is modified by removing characters "$C *$" (i) from the first two columns of the common statement (ii) from the second statement after label 105 and (iii) from the four statements following label 159 and also removing the immediately following statement at label 160. This modification has the effect of adding about $1\frac{1}{2}$ per cent to the c.p.u. time when calling *RCONT2*.

It should be noted that the decomposition (2) of P into the product of $(r-1)(c-1)$ probabilities is different from that proposed by Lancaster (1949) and discussed by Cox and Plackett (1980).

REFERENCES

AGRESTI, A., WACKERLY, D. and BOYETT, J. M. (1979). Exact conditional test for cross-classifications: approximation of attained significance levels. *Psychometrika*, **44**, 75–83.

BOYETT, J. M. (1979) Algorithm AS 144. Random $R \times C$ tables with given row and column totals. *Appl. Statist.*, **28**, 329–332.

COX, M. A. A. and PLACKETT, R. L. (1980). Small samples in contingency tables. *Biometrika*, **67**, 1–13.

LANCASTER, H. O. (1949). The derivation and partition of χ^2 in certain discrete distributions. *Biometrika*, **36**, 117–129.

```
      SUBROUTINE RCONT2(NROW, NCOL, NROWT, NCOLT, JWORK,
     *  MATRIX, KEY, IFAULT)
C
C        ALGORITHM AS 159 APPL. STATIST. (1981) VOL.30, NO.1
C
C        GENERATE RANDOM TWO-WAY TABLE WITH GIVEN MARGINAL TOTALS
C
      DIMENSION NROWT(NROW), NCOLT(NCOL), MATRIX(NROW, NCOL)
      DIMENSION JWORK(NCOL)
      REAL FACT(5001), DUMMY
      LOGICAL KEY
      LOGICAL LSP, LSM
      COMMON /B/ NTOTAL, NROWM, NCOLM, FACT
C
C*      COMMON /TEMPRY/ HOP
C
      DATA MAXTOT /5000/
C
      IFAULT = 0
      IF (KEY) GOTO 103
C
C        SET KEY FOR SUBSEQUENT CALLS
C
      KEY = .TRUE.
C
C        CHECK FOR FAULTS AND PREPARE FOR FUTURE CALLS
C
      IF (NROW .LE. 1) GOTO 212
      IF (NCOL .LE. 1) GOTO 213
      NROWM = NROW - 1
      NCOLM = NCOL - 1
      DO 100 I = 1, NROW
      IF (NROWT(I) .LE. 0) GOTO 214
  100 CONTINUE
      NTOTAL = 0
      DO 101 J = 1, NCOL
      IF (NCOLT(J) .LE. 0) GOTO 215
      NTOTAL = NTOTAL + NCOLT(J)
  101 CONTINUE
      IF (NTOTAL .GT. MAXTOT) GOTO 216
C
C        CALCULATE LOG-FACTORIALS
C
      X = 0.0
      FACT(1) = 0.0
```

```
      DO 102 I = 1, NTOTAL
      X = X + ALOG(FLOAT(I))
      FACT(I + 1) = X
  102 CONTINUE
C
C         ----------------------
C         CONSTRUCT RANDOM MATRIX
C         ----------------------
C
  103 DO 105 J = 1, NCOLM
  105 JWORK(J) = NCOLT(J)
      JC = NTOTAL
C
C*      HOP = 1.0
C
      DO 190 L = 1, NROWM
      NROWTL = NROWT(L)
      IA = NROWTL
      IC = JC
      JC = JC - NROWTL
      DO 180 M = 1, NCOLM
      ID = JWORK(M)
      IE = IC
      IC = IC - ID
      IB = IE - IA
      II = IB - ID
C
C         TEST FOR ZERO ENTRIES IN MATRIX
C
      IF (IE .NE. 0) GOTO 130
      DO 121 J = M, NCOL
  121 MATRIX(L, J) = 0
      GOTO 190
C
C         GENERATE PSEUDO-RANDOM NUMBER
C
  130 RAND = G05AAF(DUMMY)
C
C         COMPUTE CONDITIONAL EXPECTED VALUE OF MATRIX(L, M)
C
  131 NLM = FLOAT(IA * ID) / FLOAT(IE) + 0.5
      IAP = IA + 1
      IDP = ID + 1
      IGP = IDP - NLM
      IHP = IAP - NLM
      NLMP = NLM + 1
      IIP = II + NLMP
      X = EXP(FACT(IAP) + FACT(IB + 1) + FACT(IC + 1) + FACT(IDP) -
     *  FACT(IE + 1) - FACT(NLMP) - FACT(IGP) - FACT(IHP) - FACT(IIP))
      IF (X .GE. RAND) GOTO 160
      SUMPRB = X
      Y - X
      NLL = NLM
      LSP = .FALSE.
      LSM = .FALSE.
C
C         INCREMENT ENTRY IN ROW L, COLUMN M
C
  140 J = (ID - NLM) * (IA - NLM)
      IF (J .EQ. 0) GOTO 156
      NLM = NLM + 1
      X = X * FLOAT(J) / FLOAT(NLM * (II + NLM))
      SUMPRB = SUMPRB + X
      IF (SUMPRB .GE. RAND) GOTO 160
  150 IF (LSM) GOTO 155
C
C         DECREMENT ENTRY IN ROW L, COLUMN M
C
      J = NLL * (II + NLL)
      IF (J .EQ. 0) GOTO 154
      NLL = NLL - 1
      Y = Y * FLOAT(J) / FLOAT((ID - NLL) * (IA - NLL))
```

```
      SUMPRB = SUMPRB + Y
      IF (SUMPRB .GE. RAND) GOTO 159
      IF (.NOT.LSP) GOTO 140
      GOTO 150
  154 LSM = .TRUE.
  155 IF (.NOT.LSP) GOTO 140
      RAND = SUMPRB * G05AAF(DUMMY)
      GOTO 131
  156 LSP = .TRUE.
      GOTO 150
  159 NLM = NLL
C
C*        HOP = HOP * Y
C*        GOTO161
C*160     HOP = HOP * X
C*161     MATRIX(L, M) = NLM
C
  160 MATRIX(L, M) = NLM
      IA = IA - NLM
      JWORK(M) = JWORK(M) - NLM
  180 CONTINUE
      MATRIX(L, NCOL) = IA
  190 CONTINUE
C
C          COMPUTE ENTRIES IN LAST ROW OF MATRIX
C
      DO 192 M = 1, NCOLM
  192 MATRIX(NROW, M) = JWORK(M)
      MATRIX(NROW, NCOL) = IB - MATRIX(NROW, NCOLM)
      RETURN
C
C          SET FAULTS
C
  212 IFAULT = 1
      RETURN
  213 IFAULT = 2
      RETURN
  214 IFAULT = 3
      RETURN
  215 IFAULT = 4
      RETURN
  216 IFAULT = 5
      RETURN
      END
```

Algorithm AS 160

Partial and Marginal Association in Multidimensional Contingency Tables

By Edward D. Lustbader and Robert K. Stodola

The Institute for Cancer Research, The Fox Chase Cancer Center, Philadelphia, PA 19111, USA

Keywords: LOG-LINEAR MODEL; INTERACTION; VARIABLE SELECTION; DISCRETE MULTIVARIATE DATA

Language

Fortran 66

Description

Brown (1976) described a method for screening multidimensional contingency tables for significant interactions among the variables. Two tests, marginal and partial association, were